AD-A100 444 ANALYTICS INC WILLOW GROVE PA F/G 5/8
THE HUMAN OPERATOR SIMULATOR. VOLUME VIII. APPLICATIONS TO ASSE--ETC(U)
JUN 77 F A GLENN, M I STRIEB, R J WHERRY N62269-77-C-0109
UNCLASSIFIED TR-1233-A NL

1 OF 1
AD
A100444

END
DATE
FILMED
7-81
DTIC

233-A

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Technical Report 1233-A
Contract No. N62269-77-C-0109

LEVEL II

THE HUMAN OPERATOR SIMULATOR
VOLUME VIII

APPLICATIONS TO ASSESSMENT OF OPERATOR LOADING

Floyd A. Glenn
Melvin I. Strieb
Robert J. Wherry, Jr.

Analytics
2500 Maryland Road
Willow Grove, Pa. 19090

JUN 22 1981

D

14 June 1977

Final Report

Prepared for

Crew Systems Department
Naval Air Development Center
Warminster, Pa. 18974

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This study was undertaken as part of a continuing series of studies designed to validate the Human Operator Simulator (HOS). HOS is intended to be used as a system evaluation tool that can be applied to a detailed system design prior to the development of a hardware prototype. Consequently, an assessment of its ability to model operator workload problems realistically was considered to be of prime importance. The study was designed to show that HOS, a general-purpose model of human performance, can accurately simulate the kinds of complex interactions between competing task demands experienced by a human operator performing a complex mission.

Other analytic approaches to the modeling of task workload have been developed (e.g., Siegel and Wolf, 1962; Murphy and Gurman, 1972; Linton, Jahns, and Chatelier, 1977) but they all require the user to supply explicit estimates of task demands in terms of processing time and channel requirements. HOS, on the other hand, requires that the user supply only fairly objective information about hardware system configuration and dynamics, operator size and general performance characteristics, and sequences of task operations.

In this study, an operator performing both a primary pursuit tracking task and a "neutral" secondary task was simulated. The tracking task was adopted as the primary task because of the following:

1. It is commonly encountered by operators of modern control systems.

2. Its continuous nature ensures that it will be sensitive to interruptions by a secondary task.

3. It permits definitions of continuously variable performance measures.

The secondary task is neutral in the sense that the operator "does nothing," thereby avoiding the task interactions that occur when two tasks compete for processing channels. The secondary task was an externally scheduled interruption of the tracking task. Experimentally, this could be achieved by, for example, blanking out the display so that an operator engaged in visually monitored pursuit tracking could not see the display for a specified length of time. The study examined how primary task performance was influenced by the schedule of interruptions and other parameters that affect the difficulty of the tracking task (i.e., signal characteristics and the operator's criterion of acceptable performance).

2. MODELS FOR MANUAL PURSUIT TRACKING

While a simple model could be constructed that would enable HOS to perform a simple tracking function, our objective was to devise a model that would embody the kind of adaptive characteristics that human operators possess. In this section, we will discuss the rationale behind the particular model that was used, the model itself, and the simulation studies that were performed to determine some general performance features of the model. Section 3 discusses studies in which the completed tracking model was used with a variety of interruption schedules to address some basic workload measurement issues.

2.1 Control Theory Models

Most investigators who have modeled tracking behavior have adopted the "black box" approach characteristic of mathematical control theory. These control theory models have resulted in some significant insights into the organization of skilled human performance. They can achieve exceptionally accurate predictions of performance on most tracking tasks and can account for the systematic changes in performance when some interesting task factors are varied (e.g., the dynamics of the control-display relationship). However, they generally explicitly disregard the operator's internal processing structure, and often fail to provide a sufficiently detailed description of performance components to enable reliable predictions of tracking performance to be made in many situations of interest. For example, they do not permit the identification of tasks that could be performed concurrently without compromising performance on the tracking task.

In general, the best tracking model would be one whose parameter values could be determined completely from observable characteristics of the tracking apparatus and the operator's basic performance characteristics (e.g., reaction time, fatigue level). Unfortunately, control theory models generally do not admit these possibilities. Rather, the control theory approach fits a general model to a specific task situation by estimating all free parameters in order to best fit the model to actual performance. Such a modeling strategy is clearly inapplicable to the prediction of operator performance with a proposed hardware system before a prototype system can be constructed, expecially if the proposed system is very different from other systems for which performance data is available.

2.2 Information Processing

A number of investigators have studied the cognitive and motor components of tracking performance and some fairly comprehensive conceptual models have been advanced (e.g., Crossman, 1960). But, to date, there has been a singular absence in the published literature of detailed models on the operation of the processing components that contribute to tracking performance. The two factors that are probably most responsible for this are:

(1) Most human information processing operations are generally understood as discrete events (e.g., detection, decision, response), while tracking performance is apparently continuous.

(2) People are generally recognized to be extremely flexible information processors.

The problem raised by the apparent continuity of tracking performance in many situations and the presumed discreteness of human performance components, can be resolved by the modeler in a number of ways. One way is to question the continuous nature of tracking performance on theoretical or empirical grounds. Hick (1948), for example, has argued that the refractory period of neural processes must force the human operator to act as a discontinuous controller. Indeed, frequent, abrupt control movements have been observed even when the tracked signal is slow and fluid (North, Lomnicki, and Zaremba, 1948). Some psychologists have suggested that all human perceptual processing can occur only as quantitative events that are paced by a rapid biological cycle (Stroud, 1955; Kristofferson, 1967). Though compelling evidence against this theory of the "psychological moment" has been raised by experiments reported by Allport (1968) and Baron (1971), there are still clear limitations in the rates at which perceptual and motor events can proceed. Fitts and Posner (1967) suggest that the number of repetitive stimuli that can be perceptually distinguished and the frequency at which repetitive motor processes can be executed are both limited to about ten events per second. Of course, people are quite capable of executing complex continuous movements that last for many seconds, but even so, it is possible that such movements are composed completely of discrete processes. Since the perception and decision operations that must constitute the basis of human tracking performance are generally considered to be discrete processes, it is natural to use a discrete model to describe tracking.

The second point recognizes that different individuals will use very different processing strategies on the same task and even a single individual may use very different strategies on two tracking tasks that appear to differ only slightly. This tends to frustrate attempts to model behavior in tasks like tracking where the operator may choose from many different processing strategies. People probably try out several different strategies before they adopt a stable strategy, and even that final strategy may undergo frequent modifications in response to feedback. But since the possible strategies must be limited by the type of information available to the operator and by the operator's ability to interpret that information, there are only a few possibilities that require consideration.

2.3 The HOS Tracking Model

2.3.1 Rationale

In our model, we attempted to identify the information that could be available in a typical tracking task and the factors that

determine how that information will be used. This was then formulated into a model of how the information is processed to effect control movements. In addition to providing an understanding of tracking performance, it was expected that this research would provide valuable information for use in a general-purpose model of human performance. Our approach was to start with a basic model, identify the deficiencies of that model, and then modify it accordingly.

The model was developed for a pursuit tracking task in which a single display indicated the values of both the signal and the track. The value of the track was controlled by the position of a rotary knob. Our basic model assumed that the operator repeatedly executed a basic adjustment cycle consisting of:

(1) Reading the value of the signal.

(2) Reading the value of the track.

(3) Computing the error (i.e., the difference between the signal and track values).

(4a) Terminating the adjustment cycle if the error was less than a tolerance limit, or

(4b) Computing the amount of knob turn required to make the track equal to the signal (using the precise gain factor that relates knob movements to changes in the track value), and making the appropriate change if the error is greater than the tolerance limit.

Since this model makes no provision for learning, it can be considered to be a description of the stable performance of a trained operator.

Even ignoring the fact that details, such as the speed and accuracy of the component processes, have to be specified, it is clear that inadequacies exist in the model. For example, the model assumes that the operator knows the precise gain factor that relates changes in the knob's value to changes in the value of the track. While this assumption would not necessarily result in inaccurate performance predictions, it does compromise both the face validity and the generality of the model. Moreover, the model assumes that the operator does not rely on his experience based on previous adjustment cycles to anticipate future values of the signal. This assumption would preclude the occurrence of tracking overshoots at signal inflections and tracking lags shorter than the operator's reaction time. Since real operators do frequently overshoot the maxima and minima of the signal and, for some signals, exhibit very short lags, modifications to the basic model were clearly necessary. The modifications that were implemented provide the model with the capability to learn both a signal extrapolation time and an estimate for the system gain factor.

2.3.2 Notation

In discussing the modified tracking model, the following notation will be used:

S_n = the operator's estimate of the signal value on the n^{th} adjustment cycle

S_n* = the operator's extrapolated estimate of the signal value on the n^{th} cycle

t_n = the extrapolation time used by the operator on the n^{th} cycle

T_n = the operator's estimate of the track value on the n^{th} cycle

g_n = the operator's estimate of the gain factor on the n^{th} cycle

Δ = the operator's tolerance for being "on target" (assumed to be constant)

L_t = the extrapolation time learning function

L_g = the gain factor learning function

2.3.3 Sequence of Operations

The model assumes that on any adjustment cycle, the operator performs the following sequence of operations:

(1) Reads the value of the signal, S_n

(2) Computes the new values for the extrapolation time, $t_n = L_t\ (t_{n-1}, \ldots)$

(3) Computes the extrapolated signal value, S_n*

(4) Reads the value of the track, T_n

(5a) Terminates the adjustment cycle if $|S_n* - T_n| \leq \Delta$, or

(5b) Computes the new value for the gain factor, $g_n = L_g\ (g_{n-1} \ldots)$, the desired knob turn, and makes the appropriate adjustment if $|S_n* - T_n| > \Delta$.

The speeds and accuracies for the display reading and control manipulation operations are automatically assigned by HOS micro-models. The HOS micro-model for display reading, described in detail in Volume VII of this series, models the reading process by a sequence of discrete

micro-absorptions that are terminated either when successive estimates differ by a small enough amount that the operator is considered to have "learned" the value, or when a time limit for the reading process is exceeded. The estimated value of the display on each micro-absorption is obtained by averaging the actual value of the display with a value extrapolated linearly from the previous two micro-absorption estimates. The time consumed by each micro-absorption is defined as a constant cost plus an additional amount dependent on the dissimilarity between the estimated values of the current and previous micro-absorptions. The effects of this display reading model in the tracking simulation studies are:

- Display values are always read to within a narrowly defined additive tolerance of the actual value
- The time for each reading varies between about .02 seconds and about .2 seconds
- The longest reading times are associated with the most extremely nonlinear changes in signal values.

The HOS control manipulation micro-model causes the adjustments to be made to precisely the intended setting. Manipulations require an amount of time that is a linear function of the magnitude of the adjustment. The time consumed in turning a knob is given by the function

time (in seconds) = .0029A + .0982

where A is the angle in degrees through which the knob is turned.

All operations, other than display readings and knob adjustments (i.e., all computations and the one comparison), were assigned zero time costs.

2.3.4 The Learning Functions

The two learning functions were defined so that the amount of information that the operator had to store and use was kept to a minimum. In order to be able to extrapolate the signal value, it was necessary to assume that the operator "knew" both the current and previous estimates of the signal and the times when those estimates were obtained. He also had to know the current and previous values of the track, the current working values for the gain factor and extrapolation time, and a running average for the absolute tracking error.

A gain learning function must be sensitive to the effect of the most recent control manipulation, increasing the estimated gain factor if that manipulation was too large and decreasing the estimated gain if the manipulation was too small. An extrapolation time learning function must be sensitive to the extent to which the track is tending to lead or lag behind the signal, increasing the extrapolation time if the track is lagging behind the signal and decreasing the extrapolation time if the

track is leading the signal. A set of learning functions that have these properties is shown in Table 1. The parameters ϕ, θ, the initial learning rates for extrapolation and gain, and δ, the relative size of random changes to the gain, are input constants. ϕ, θ, and δ are defined on the range of 0 to 1 and are considered to be of approximately the same magnitude.

These learning functions imply that the effective learning rates on any adjustment cycle (ϕ_n and θ_n) are the product of the base learning rates and the ratio of the estimated error (E_n) to the average absolute error (A_n) on any cycle. Thus, relatively small changes are made to both the extrapolation and gain values when the tracking error is relatively small and relatively large changes are made when the tracking error is large.

The extrapolation learning function increments or decrements the old extrapolation value by the product of the old extrapolation value and the effective learning rate, incrementing when the track lags behind the signal and decrementing when the track leads the signal.

The gain learning function determines the new gain by taking a weighted average of the old gain and an estimate for the "ideal" gain, the weighting factor being the effective learning rate. The ideal gain calculation assumes that the signal will not change appreciably over a single adjustment cycle. Then, if the actual adjustment on the $n\text{-}1^{st}$ cycle was optimal, the error on the n^{th} cycle would be zero. The actual adjustment made on the $n\text{-}1^{st}$ cycle, α, is given by

$$\alpha = \frac{S_n^* - T_n}{g_{n-1}} \cong - \frac{E_{n-1}}{g_{n-1}}$$

and the adjustment that would have been optimal for that cycle, α^*, is given by

$$\alpha = - \frac{E_{n-1}}{g^*}$$

where g^* is the corresponding optimal gain.

Since the actual adjustment changed the tracking error from E_{n-1} to E_n and the optimal adjustment would have changed the error from E_{n-1} to 0, the assumption that the control is linear implies that

$$\frac{\alpha^*}{\alpha} = \frac{E_{n-1}}{E_{n-1} - E_n} \cong \frac{g_{n-1}}{g^*}$$

TABLE 1. LEARNING FUNCTIONS FOR PURSUIT TRACKING

I. For running average of absolute error:

$$A_n = .9\,A_{n-1} + .1\,|E_n|$$

II. For extrapolation:

$$t_n = \begin{cases} (1 + \phi_n)\,t_{n-1} & \text{if } S_n \le S_{n-1} \\ (1 - \phi_n)\,t_{n-1} & \text{if } S_n > S_{n-1} \end{cases}$$

$$\text{where } \phi_n = \max\left(\min\left(\frac{\phi E_n}{A_n}, .2\right), -.2\right)$$

III. For gain:

$$g_n = [\theta_n f_n + (1 - \theta_n)]\,g_{n-1}$$

$$\text{where } \theta_n = \min\left(.9, \frac{\theta|E_n|}{A_n}\right)$$

and

$$f_n = \begin{cases} \max\ \min\left(1 - \frac{E_n}{E_{n-1}}, 2\right), .5) & \text{if } E_{n-1} = 0 \text{ and } \frac{E_n}{E_{n-1}} \le 1 \\ \left.\begin{array}{ll} 1 + \delta & \text{with probability } \tfrac{1}{2} \\ \frac{1}{1 + \delta} & \text{with probability } \tfrac{1}{2} \end{array}\right\} & \text{if } E_{n-1} = 0 \text{ or } \frac{E_n}{E_{n-1}} > 1 \end{cases}$$

where $E_n = T_n - S_n$ is the estimated error on the n^{th} adjustment cycle

and that

$$g^* \cong \frac{E_{n-1} - E_n}{E_{n-1}} \; g_{n-1}.$$

If, however, the adjustment causes an increase in the error, i.e., if

$$\frac{E_n}{E_{n-1}} > 1,$$

then this formula implies that g* and g_{n-1} must have different signs, corresponding to a change in the estimated directional relationship between the control and the track value. Since such sign changes would result in a highly unstable learning process, it was assumed that the operator knows the directional correspondence between the control and the display. Therefore, the model permits only positive estimates for the gain. When

$$\frac{E_n}{E_{n-1}} > 1$$

or when

$$E_{n-1} = 0,$$

a new value for the ideal gain was considered to be indeterminable. Since such indeterminate cases represent conditions of deteriorating tracking performance, the new gain value for that cycle was randomly defined as a fixed increment or decrement from the old gain, each change occurring with a probability of .5.

For both gain learning and extrapolation learning, the maximum change made to either the gain or extrapolation time on a single cycle is indicated by the truncation operations shown in Table 1. These limitations keep the gain learning from being dominated by the special cases

$$E_n << 0 < E_{n-1}$$

and

$$E_{n-1} < 0 << E_{n-1}$$

and extrapolation learning by the special case:

$$|E_n| >> \frac{A_n}{\emptyset}.$$

2.4 Parametric Behavior of the HOS Tracking Model

Despite its similarity to the linear learning models that have been extensively studied by mathematical psychologists (Bush and Mosteller, 1956; Norman, 1968), the tracking model is sufficiently different from those models so that very different methods of analysis are necessary. In particular, other mathematical models for learning have dealt principally with the learning of probabilities and probability distributions rather than with the learning of strategic parameters like system gain and extrapolation time. Also, other learning models were developed for situations in which learning occurred on discrete trials for which the experimenter-controlled events (e.g., stimuli and rewards) were scheduled independently. For the tracking situation, the stimulus is continuous and successive learning events are consequently correlated. Further complicating the analysis of the tracking model is the fact that the learning functions are discontinuous unlike the learning functions used in other models. And even if none of these features had precluded an analysis of the model, the fact that it uses two interacting learning functions would have made the model sufficiently intractable analytically. Thus, the only viable method for studying the behavior of the model is to examine the results of dynamic simulations for a variety of different model parameters and different types of signal.

2.4.1 Measures of Performance

In evaluating the behavior of the tracking model, it is important to consider both the accuracy of tracking performance and the "success" of the two learning processes. Two different tracking performance measures, root-mean-squared-error (RMSE) and percent-time-on-target (PTOT) were used to assess tracking accuracy, since it would have been impossible to predict which measure would be most sensitive to the performance differences.* The success of the learning process was evaluated based on its stability (i.e., variance and regression over time) and central tendency.

Each tracking trial lasted for 60 seconds of simulated time and started with an initial gain value of 200 display units per degree of knob turn (the actual gain of the simulated system) and an initial extrapolation time of .1 second (somewhat less than the adjustment cycle time of the simulated operator). Performance data were collected only for the last 50 seconds of each trial. Trials were conducted for several values of the model parameters θ, δ, and $\emptyset$, the objective being to get a general idea of how the parameters governed model performance. In all cases, the signal was sinusoidal with an offset of 10,000 units and an amplitude of either 4000 or 8000 units. For each combination of model parameters, the first tracking trial was conducted with a signal frequency of .1 cycles/second and subsequent trials with the same

---

* For present purposes, PTOT is arbitrarily defined as the percentage of time that the track is within 400 units of the signal.

parameters used signal frequencies increased by a factor of $\sqrt{2}$ until performance was degraded to an uninteresting level.

2.4.2 Simulation Studies

The results of these parametric runs are shown in Table 2. Section A of the table demonstrates the baseline performance -- without learning, but with a precisely correct gain value and a fairly small extrapolation time. Note that the two overall performance measures, RMSE and PTOT, appear to be about equally sensitive to each increment in signal frequency. In Section B of Table 2, only the gain learning features of the model were being exercised and, in Section C, only the extrapolation time learning component was active. Sections D and E indicate how the model behaves when both the gain and extrapolation time learning features are exercised together. In Section E, the signal had twice the amplitude of the signal used in Section D. Finally, Section F displays model performance when both gain and extrapolation time are learned but at rates that are twice as large as those used in other cases. Data for only three different signal frequency values are presented in Section F because during the simulation with the fourth frequency (.2828 cycles per second), an unrealistically large estimate for the extrapolation time resulted in a knob-turn that was not completed during the span of the simulation.

Figure 1 shows how the learned gain value varies with signal frequency. From Table 2 it can be seen that the points with the largest gain values correspond to trials with large regression coefficients, so the results are, in fact, underestimates of any ultimately stable gain value that might (or might not) be attained. When only gain learning is present (case b), the learned gain is fairly accurate (with respect to actual system gain) at the lowest frequency. It stabilizes at a value about 35% above the actual system gain at the midrange of frequencies from .2 to .4 cycles/second and becomes unstable above .5656 cycles/second. When both gain and extrapolation time are being learned (case d), similar stable gain values are learned for low signal frequencies, but the model appears to become unstable at a lower frequency (by .4 cycles/second) than in case b. Curiously, when the signal amplitude is doubled and both learning processes are operative (case e), a fairly high gain value is learned at the lowest signal frequency, while lower, more accurate values are learned for slightly higher frequencies, with instability occurring at roughly the same point as at the smaller signal amplitude. When the learning rates for both processes are doubled (case f), the learned gain appears to be virtually the same as when the smaller learning rates are used.

Figure 2 shows how the learned extrapolation time varies with signal frequency. Again, the regression coefficients in Table 2, indicate that the larger extrapolation times correspond to the means of values that increase systematically with simulation time. All model cases show basically the same relationship between learned extrapolation time and signal frequency when the signal amplitude is constant. At the lowest frequency (.1 cycles/second) a stable extrapolation time of

TABLE 2. RESULTS OF PARAMETRIC RUNS

					GAIN			EXTRAPOLATION TIME				
Θ	δ	Φ	SIGNAL AMP	SIGNAL FREQ(HZ)	MEAN	STAND. DEV.	REGRESS. COEF.	MEAN	STAND. DEV.	REGRESS. COEF.	RMSE	PTOT (SIZE = 400)
0	0	0	4000	.1	200.00	--	--	.100	--	--	265.0	88.5
0	0	0	4000	.1414	200.00	--	--	.100	--	--	354.0	71.2
0	0	0	4000	.2000	200.00	--	--	.100	--	--	639.0	42.5
0	0	0	4000	.2828	200.00	--	--	.100	--	--	1264.9	22.4
0	0	0	4000	.4000	200.00	--	--	.100	--	--	2461.0	5.4
0	0	0	4000	.5657	200.00	--	--	.100	--	--	3873.6	4.6
.05	.02	0	4000	.1000	175.56	10.00	-.041	.100	--	--	262.1	89.2
.05	.02	0	4000	.1414	220.50	15.66	-.237	.100	--	--	385.3	62.3
.05	.02	0	4000	.2000	277.22	24.40	1.191	.100	--	--	782.9	28.5
.05	.02	0	4000	.2828	279.89	16.84	.015	.100	--	--	1425.3	14.5
.05	.02	0	4000	.4000	265.83	9.72	.162	.100	--	--	2878.9	5.3
.05	.02	0	4000	.5657	320.27	54.53	2.886	.100	--	--	3349.7	6.0
.05	.02	0	4000	.8000	642.60	200.61	11.000	.100	--	--	3509.4	5.5
0	0	.05	4000	.1000	200.00	--	--	.224	.040	.000	169.2	99.6
0	0	.05	4000	.1414	200.00	--	--	.246	.051	.001	322.0	83.5
0	0	.05	4000	.2000	200.00	--	--	.311	.066	.000	786.3	45.2
0	0	.05	4000	.2828	200.00	--	--	.415	.086	.001	1689.1	14.6
0	0	.05	4000	.4000	200.00	--	--	4.336	3.681	.213	45125.0	1.0
0	0	.05	4000	.5657	200.00	--	--	1.419	1.105	.055	18633.3	1.4
0	0	.05	4000	.8000	200.00	--	--	.451	.199	.006	10262.7	1.9
.05	.02	.05	4000	.1000	202.95	29.11	.927	.224	.046	.000	172.7	98.7
.05	.02	.05	4000	.1414	231.24	48.95	-2.244	.262	.061	.000	301.4	84.9
.05	.02	.05	4000	.2000	359.49	44.58	1.998	.433	.097	.003	811.7	35.8
.05	.02	.05	4000	.2828	246.57	30.74	- .582	.470	.101	.002	1796.6	11.7
.05	.02	.05	4000	.4000	535.29	291.07	15.955	11.394	10.819	.600	24907.7	1.0
.05	.02	.05	4000	.5657	678.14	549.22	29.564	5.240	6.165	.336	14121.3	1.4
.05	.02	.05	8000	.1000	352.34	56.67	1.388	.433	.098	.003	686.0	46.1
.05	.02	.05	8000	.1414	244.46	35.14	- 1.284	.397	.081	.000	1005.0	31.2
.05	.02	.05	8000	.2000	194.00	42.62	- 2.162	.525	.128	.001	2522.3	11.8
.05	.02	.05	8000	.2828	195.72	24.66	- 1.182	1.054	.478	.026	11424.2	2.1
.05	.02	.05	8000	.4000	399.03	215.93	11.811	4.133	3.814	.210	27039.4	1.0
.05	.02	.05	8000	.5657	771.80	583.47	32.142	3.227	3.533	.189	15864.0	1.4
.10	.04	.10	4000	.1000	210.53	27.31	.537	.211	.068	.000	201.75	96.0
.10	.04	.10	4000	.1414	213.14	30.76	- .122	.235	.066	.000	336.2	79.3
.10	.04	.10	4000	.2000	328.23	81.15	2.600	.400	.152	.002	885.8	39.0

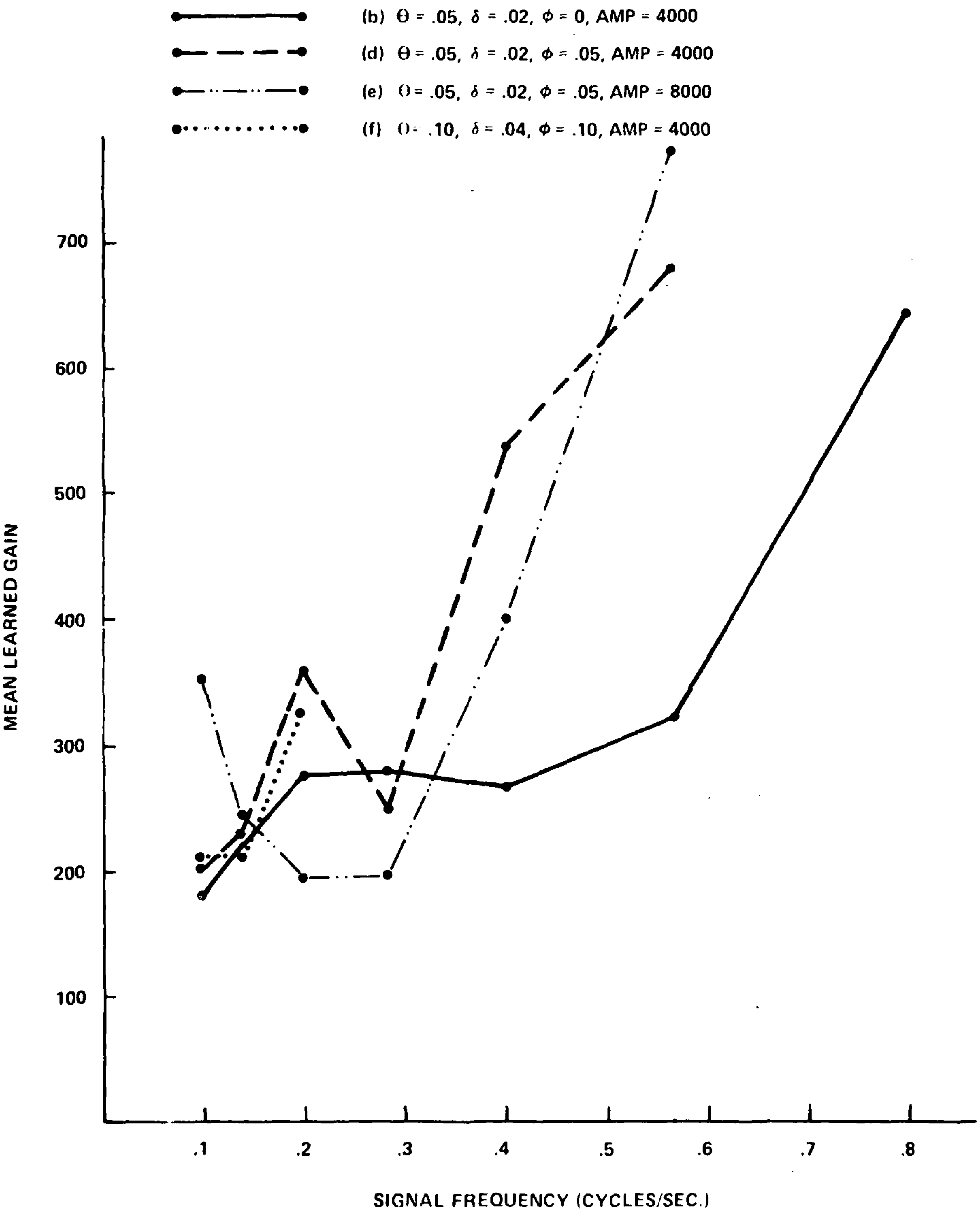


FIGURE 1. MEAN LEARNED GAIN VS. SIGNAL FREQUENCY

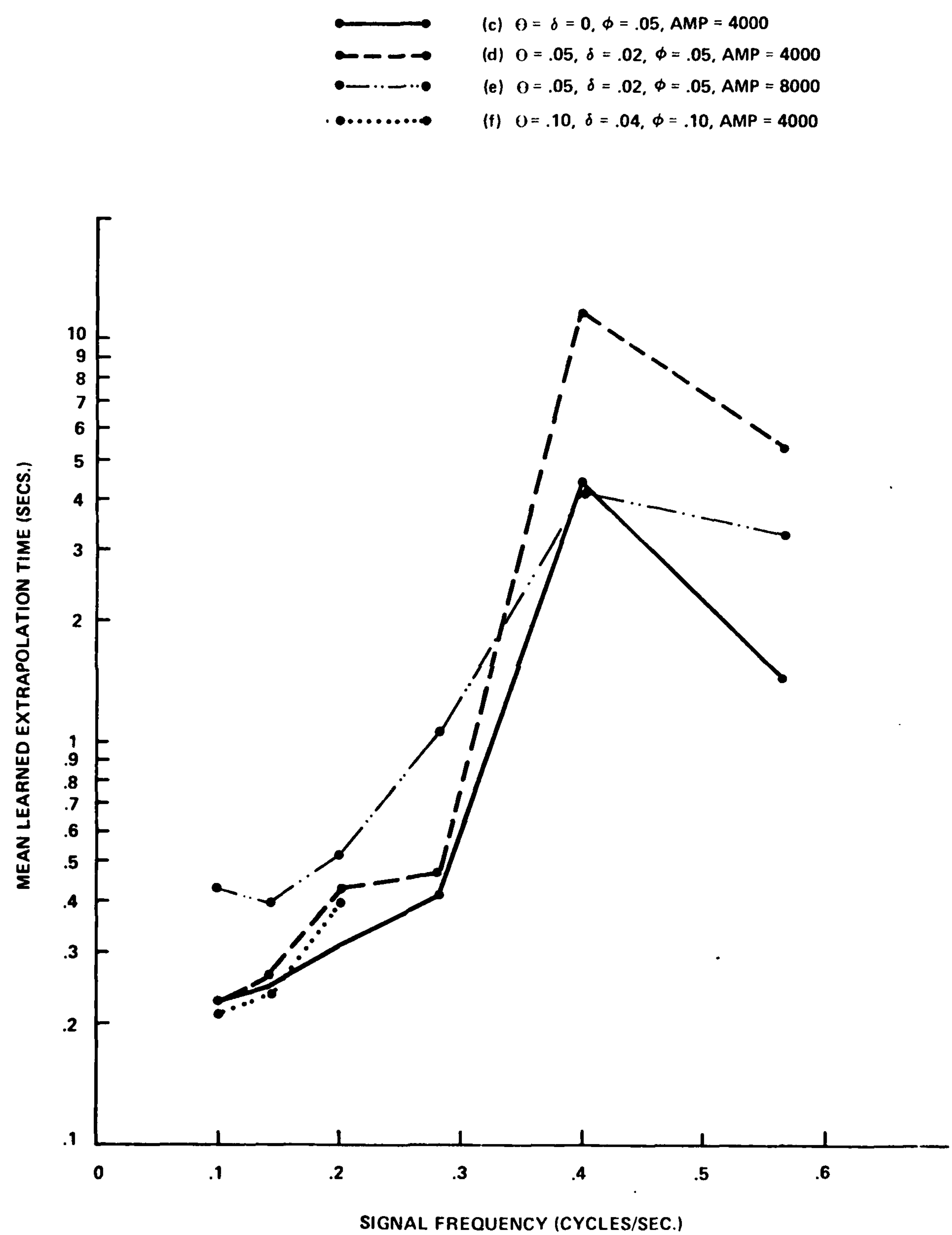


FIGURE 2. MEAN LEARNED EXTRAPOLATION TIME VS. SIGNAL FREQUENCY

about .22 seconds is learned. At higher signal frequencies, larger extrapolation times are obtained, the relationship being approximately exponential over the midrange of frequencies. The extrapolation time peaks at .4 cycles per second in all cases and is followed by a slightly lower, but still unstable, value at the next higher frequency. For the larger signal amplitude (case e), the major difference occurs at the lower signal frequencies where the extrapolation time obtained is considerably larger than for the other case.

In Figure 3, the overall tracking performance accuracy measured by RMSE is plotted as a function of signal frequency. In Figure 4, accuracy as measured by PTOT is plotted. Both plots show the same general trends, though the trends are somewhat more apparent in Figure 3 because of the scales used. Approximately the same performance is achieved when neither quantity is learned and when only the gain is learned. When the extrapolation time is learned, performance is essentially unaffected by the addition of gain learning. Whenever the extrapolation time is learned, performance is better than the corresponding baseline level (case a) at the two lowest frequencies and worse than baseline at the higher frequencies. Tracking accuracy is poorest at .4 cycles per second, the point at which the extrapolation time learning process was least stable. Not surprisingly, performance with the doubled signal amplitude is uniformly worse than with the standard amplitude signal except at the highest frequencies where performance was comparable to that obtained in the other extrapolation time learning situations.

2.4.3 Conclusions

These results indicate that the model performance is reasonable for most situations. However, some unexpected, but unmistakable, trends are evident. The fact that the learned model parameters do not stabilize in some situations is disturbing, since it is difficult to conceive of a corresponding feature in human tracking performance. However, large estimated gain values correspond to small knob adjustments. Thus, the "unstable" gain values obtained at high frequencies tend to dampen the knob turning responses, which would have been more erratic without gain learning. When the signal frequency is slow enough to permit fairly successful tracking, the gain learning model behaves quite reasonably, converging near the actual system gain.

The extrapolation time learning exhibits some suspicious characteristics with unstable learning at frequencies above .2828 cycles per second and with larger rather than smaller times being obtained as the frequencies increase. Since, however, it is not clear how people extrapolate the signal in such a situation, the trend may, in fact, constitute a valid description of a real component of human tracking behavior.

Although detailed time plots of signal and track have not been included because of their low information content, it should be noted that extrapolation learning, either alone or with gain learning, produced overshoots at signal inflections and definite compensation for the operator's tendency to lag behind the signal. Gain learning alone did

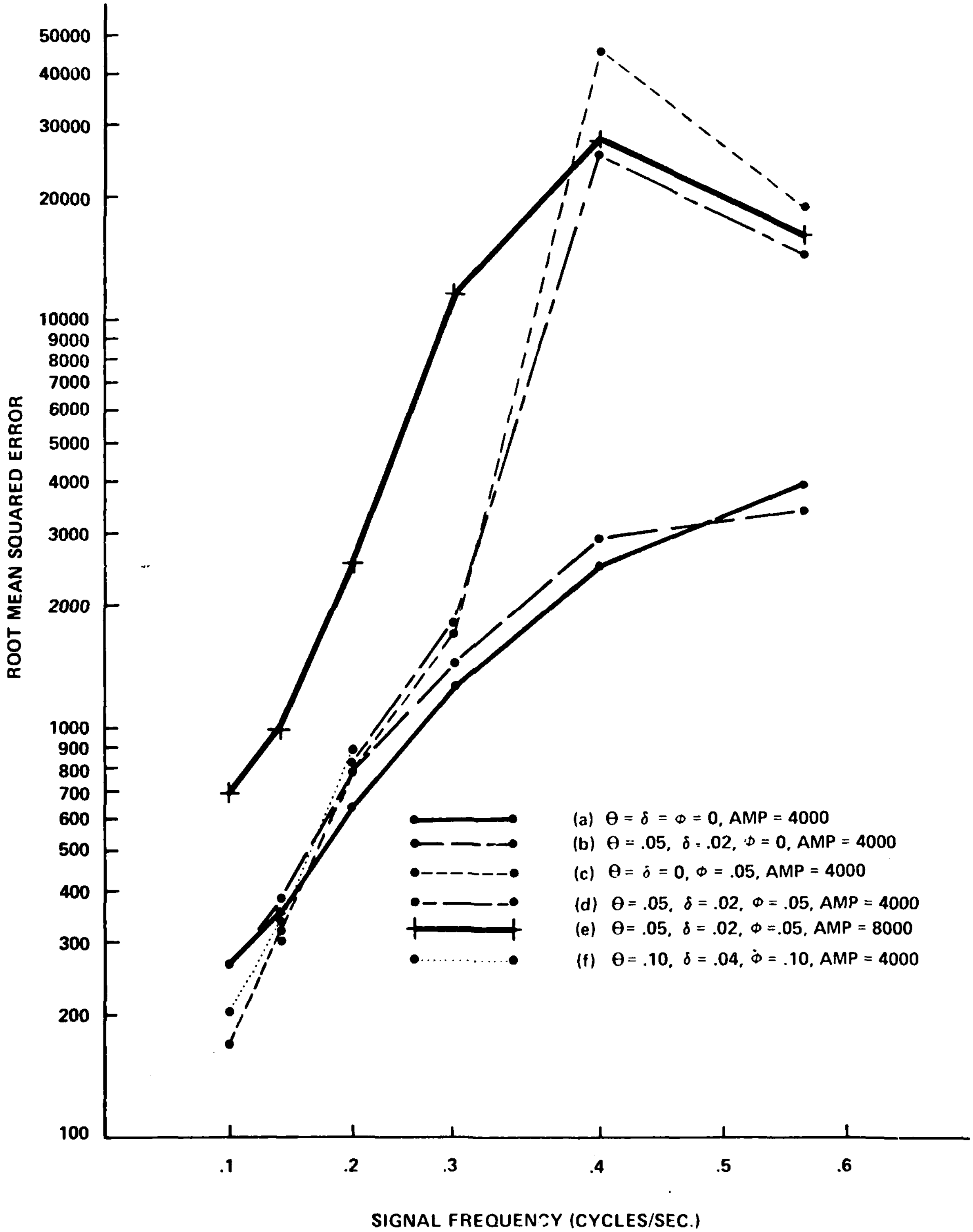


FIGURE 3. ROOT MEAN SQUARED ERROR VS. SIGNAL FREQUENCY

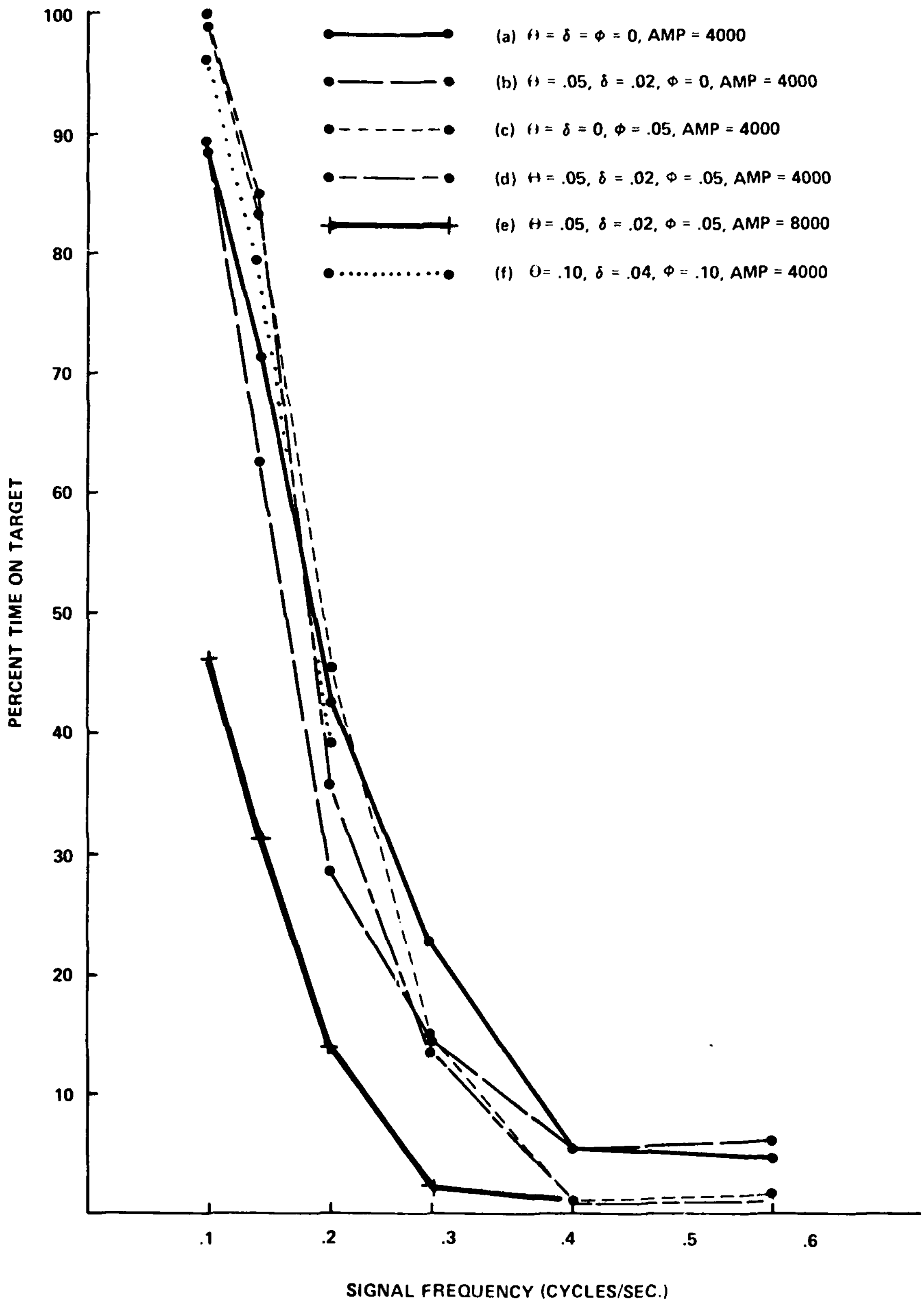


FIGURE 4. PERCENT TIME ON TARGET VS. SIGNAL FREQUENCY

not. Gain learning could have been expected to generate such features if the learned gain value were smaller than the actual system gain, but that did not happen reliably in any of the situations that were simulated.

2.4.4 Recommendations

Before further development of our model is undertaken, a systematic comparison of the model predictions with the performance of real human operators should be made. The experiments that we have simulated should be a good starting place for such a study. In addition, it would be desirable to compare both human and model performance with signals more complex than a simple sine wave. In this context, it would be valuable to study the extent to which the model could be characterized as a linear system. Such investigations will probably result in modifications to the model, especially to the extrapolation learning component, but without such studies, it is impossible to determine what changes are necessary.

3. STUDIES OF WORKLOAD

Tasks that are vulnerable to degradation by interruption could reasonably be said to impose a greater workload on the human operator. therefore, if the workload requirement of a continuous task like tracking could be characterized by a performance function that related performance when the primary task is interrupted for any given percentage of time to uninterrupted tracking performance, a major step toward defining a general workload metric would have been achieved. If, however, the performance function was dependent on the particular interruption schedule, even though the total amount of interruption time was fixed, then it would be much less useful in assessing workload. The studies to be described in this section were undertaken to determine the feasibility of a general workload metric using HOS's ability to simulate interrupted tracking tasks. While the validity of these studies are, of course, dependent upon a validation of the HOS tracking model, the studies do provide interesting insight into the workload metric issue and a solid set of performance data for eventual comparison with experimental data.

Since there is no a priori way to determine which might be most sensitive to the factors being examined, both root-mean-squared error (RMSE) and percent time-on-target (PTOT) were used as measures of performance. For PTOT, five different criteria of "on-target" were used to insure that any real effects would be detected.

The specific code that implements the tracking model and other input data required by these studies is listed in Appendix II.* Some minor modifications were made to HOS in order to simplify the required data processing. These modifications are discussed in Appendix III. In order to compute the performance measures reported in these studies, a special purpose data analysis program (called EVAL) was written. That program is described and listed in Appendix IV. Each tracking trial in the four studies to be described lasted for 220 simulated seconds. Performance data were collected only over the last 200 seconds in order to avoid transient effects. The parameters used for all of the workload studies to be reported are:

$$\theta = .05$$

$$\delta = .02$$

$$\phi = .05$$

and, except for Study 4,

$$\Delta = 800 \text{ units.}$$

---

* The code and data used for the parametric studies described in Section 2 are listed in Appendix I.

3.1 Study 1 -- The Effects of Task Difficulty and Time-Sharing on Tracking Performance

This study examined the sensitivity of the HOS tracking model to changes in signal characteristics and secondary task loading. It was expected that as the frequencies and amplitudes of the component sine waves that form the signal became faster and larger, respectively, performance should degrade. Similarly, it was expected that performance would be worse with secondary task interruptions that without them. These hypotheses were tested by requiring the simulated operator to track three different signals both with and without the secondary task for a total of six simulation trials. The three signals were the composite of three sine waves of different frequencies and amplitudes, as shown below:

	AMP	FREQ (Hz)	AMP	FREQ (Hz)	AMP	FREQ (Hz)
Signal 1	2000	1/10	1400	1/7	800	1/4
Signal 2	4000	1/20	2800	1/14	1600	1/8
Signal 3	4000	1/10	2800	1/7	1600	1/4

Thus, all three signals had the same basic shape. Signal 1 differed from Signal 3 only by having the component amplitudes reduced by a factor of two and Signal 2 differed from Signal 3 by having the component frequencies reduced by a factor of two.

The results for Study 1 are presented in Table 3. They indicate definite performance degradation when either the frequency or the amplitude of the signal increases or when secondary task interruptions occur. Nothing further can be concluded from these results about the relationship between performance and signal characteristics since only two sets of frequencies and amplitudes were used.*

The data in Table 3 clearly indicate that when examining the performance decrements caused by interruptions to the primary task, one must be aware of the implications of using a specific performance measure. For example, if one uses the differences between displayed scores as the measure of performance decrement, then, based on RMSE, performance on Signal 1 is least degraded by introducing interruptions and performance on Signal 3 is most degraded. If differences between PTOT are taken, though, exactly the opposite results are obtained. If the ratios of scores rather than the differences are used, yet another ordering is obtained and if the PTOT scores are converted to "time off target" scores still another ordering. One is forced to conclude, therefore,

---

* However, the studies described in Section 2 examined the relationship between model performance and signal characteristics without any secondary task interruptions in more detail.

TABLE 3. RESULTS OF STUDY 1

TRACKING		TASK ALTERNATION PERIOD'S DURATION				SUMMARY OF ANALYSIS OF CRITERION MEASURES						
							PERCENT TIME-ON-TARGET; SIZE:					
SIGNAL NO.	TOTAL % OF TIME	TOTAL	TRACK	DELAY	INTERNAL LIMITS	RMS ERROR	1600	800	500	300	100	CONTROL ADJUSTS
1	100.0	6.0	6.0	.0	800	455	99.4	93.7	74.7	48.9	17.5	396
1	60.0	10.0	6.0	4.0	800	1417	81.6	65.1	44.4	28.7	9.7	173
2	100.0	6.0	6.0	.0	800	562	99.9	82.7	56.3	35.8	13.2	215
2	60.0	10.0	6.0	4.0	800	2625	71.5	59.0	40.2	24.3	8.4	155
3	100.0	6.0	6.0	.0	800	910	91.7	71.5	50.4	32.8	11.3	625
3	60.0	10.0	6.0	4.0	800	3357	66.5	51.2	37.7	23.9	8.1	309

that any scaling of task difficulty according to the size of the performance decrement produced by introduction of a secondary task is arbitrary and must be used with caution, if at all.

3.2 Study 2 -- The Effect of Length of "Alternation Period" on Primary Task Performance

There are many reasons why one would expect tracking performance to vary when the percentage of interruption time is held constant and the specific time for each primary and secondary task execution is varied. For one thing, as the length of the interruption increases, the farther (on the average) the signal is likely to have drifted from the last "acquired" position. However, increasing primary task execution time gives the operator more time to make corrections once he returns to the primary task. Therefore, it is more likely that he will be able to resume satisfactory tracking of the signal. Returning to stable performance after an interruption also requires that the two component learning processes and error compensation have time to stabilize. Thus, the recovery time is likely to be dependent on the learning rates. Therefore, varying the alternation period between tasks could lead to either improved, unchanged or degraded performance depending on the choice of signal characteristics and model parameters.

Study 2 addresses these questions. A single signal was used for all trials (Signal 3 from Study 1) and six different alternation periods were examined. In each case, sixty percent of the time was spent on the primary task and forty percent on the secondary task. The length of the primary-secondary alternation cycle was varied from 2.5 seconds to 15 seconds in 2.5 second increments.

The results of this study are shown in Table 4 and plotted in Figure 5. Both RMSE and PTOT are negatively accelerated functions. All PTOT scores indicate that the best performance for the particular combination of model and signal parameters used was obtained at an alternation period of 10 seconds, although the drop-off at longer periods is very slight. It is not clear whether RMSE would continue to rise at alternation periods longer than those examined but it definitely increases over the lower part of its range and is near its maximum value at 10 seconds. Thus, the alternation period which PTOT indicates as optimal is clearly sub-optimal by the RMSE measure.

Tracking performance degrades at the beginning of an interruption and recovers after an interruption. Figure 6 shows the degradation and recovery time trajectories as measured by PTOT with a criterion tolerance of 1600 units. The envelopes marked "recovery" and "deterioration" are the envelopes of the averaged second-by-second scores for the six trials in the study. The dashed curves are "eyeball" fits based on the two recovery points (i.e., the average of the first second and the average of the second second) plotted for each alternation period. The trajectories indicate that after two seconds of time on the primary task, recovery is virtually complete. Recovery tends to be slower at the ten-second alternation period than at other periods, raising further doubts about the PTOT maximum noted above.

TABLE 4. RESULTS OF STUDY 2

TRACKING		TASK ALTERNATION PERIOD'S DURATION				SUMMARY OF ANALYSIS OF CRITERION MEASURES							
							PERCENT TIME-ON-TARGET; SIZE:						
SIGNAL NO.	TOTAL % OF TIME	TOTAL	TRACK	DELAY	INTERNAL LIMITS	RMS ERROR	1600	800	500	300	100	CONTROL ADJUSTS
3	60.0	2.5	1.5	1.0	800	3004	42.7	31.0	20.5	12.5	3.7	360
3	60.0	5.0	3.0	2.0	800	3243	59.9	42.1	27.6	16.3	4.8	301
3	60.0	7.5	4.5	3.0	800	3613	65.9	44.5	30.3	18.8	6.7	317
3	60.0	10.0	6.0	4.0	800	3357	66.5	51.2	37.7	27.9	8.1	309
3	60.0	12.5	7.5	5.0	800	3523	65.7	49.7	33.4	20.7	6.7	293
3	60.0	15.0	9.0	6.0	800	3713	65.2	49.0	34.3	21.5	7.4	313

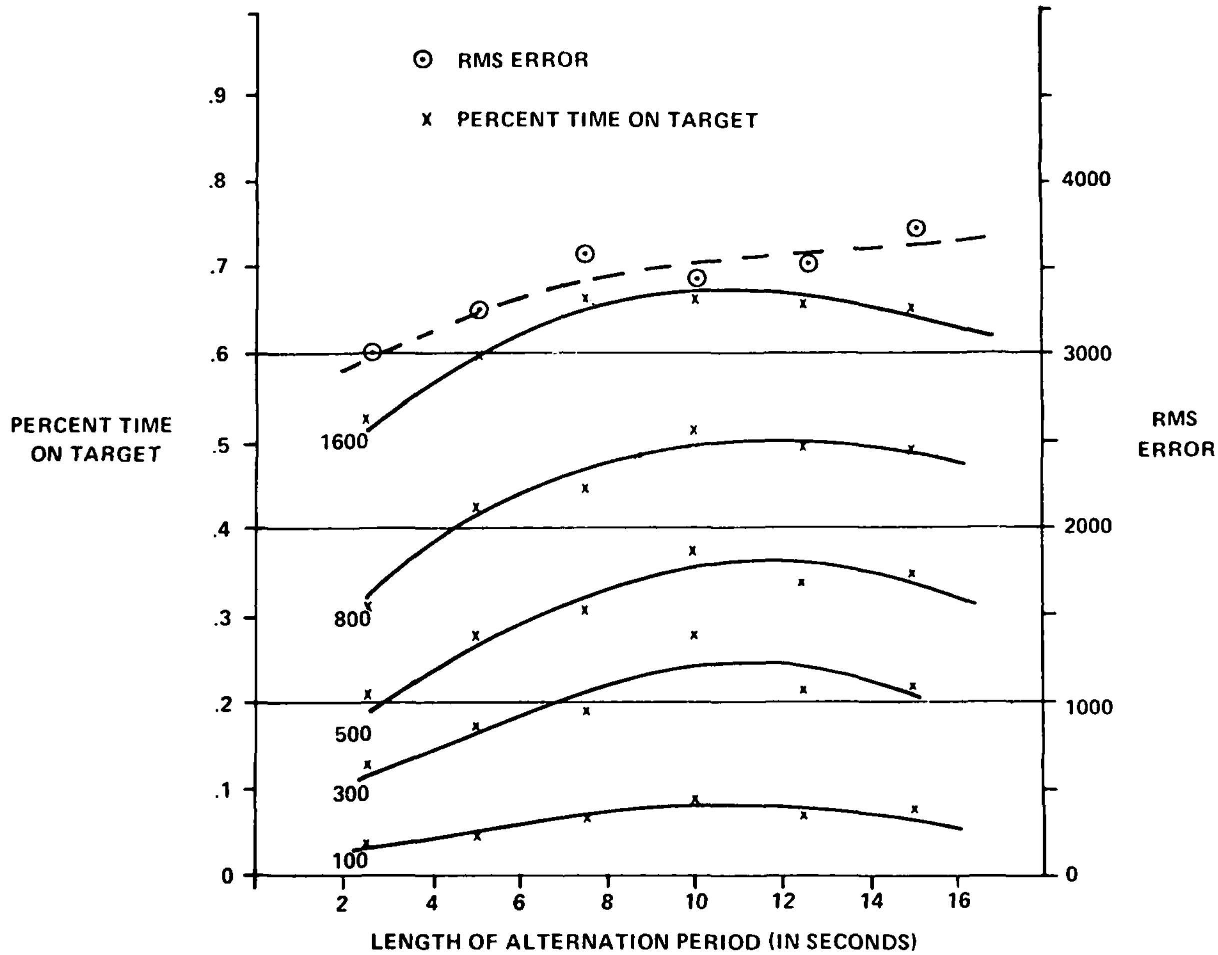


FIGURE 5. ALTERNATION PERIOD VS. RMS ERROR AND TIME ON TARGET

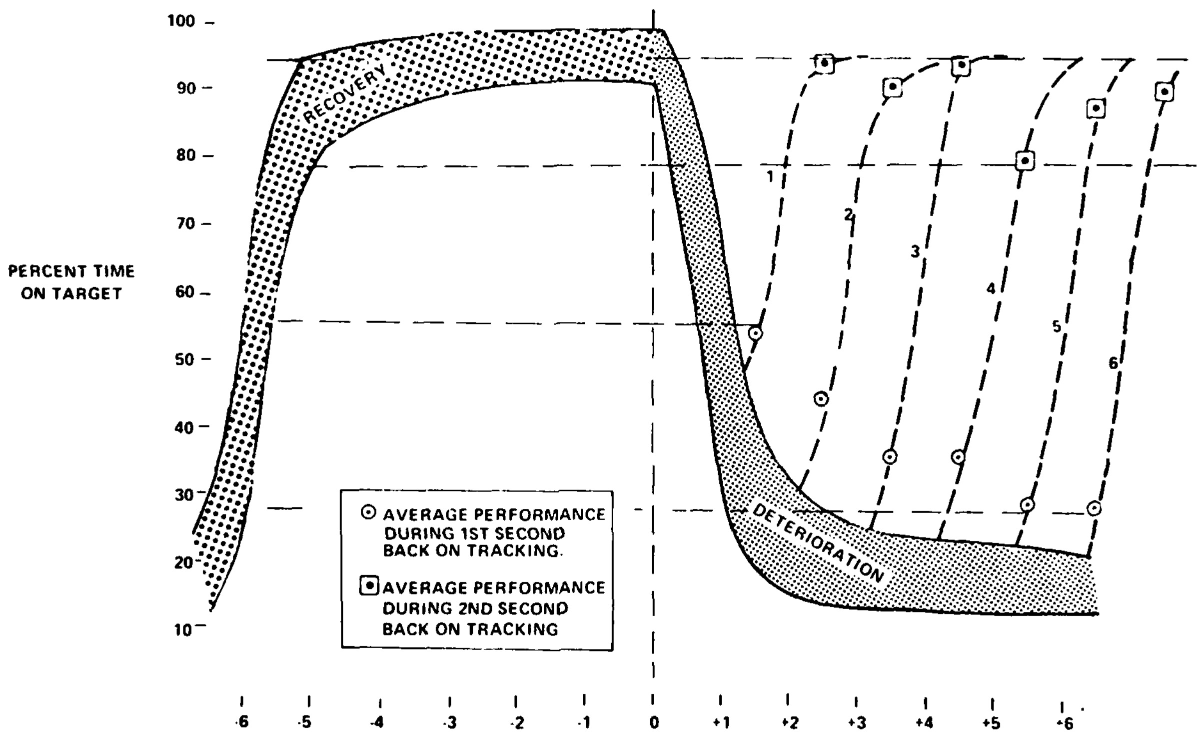


FIGURE 6. SECONDS FROM INTERRUPTION OF TRACKING VS. TIME ON TARGET

3.3 Study 3 -- The Effect of Interruption Time on Tracking Performance

If the time spent working on the primary task after each interruption is fixed and only the duration of interruptions is allowed to vary, then longer interruptions should result in poorer overall tracking performance. Study 3 examines this effect. Signal 3 of Study 1 was again used for all trials. For each trial, the simulated operator was permitted to work for six seconds on the primary task without interruption. The length of the interruptions were varied from zero to six seconds in one second increments so that the percentage of time spent on the primary task varied from 100 to 50 percent.

The results for this study are listed in Table 5 and plotted in Figure 7. The lines in Figure 7 are least-square fits. As is clear, both RMSE and PTOT vary linearly with the percentage of time on the primary task. It is interesting to note that the y-intercepts of the PTOT fits are approximately the time-on-target frequencies that would be expected if the operator spent no time on the tracking task and if the signal values were uniformly distributed over the signal range (1600 units to 18,400 units) while the track element was fixed at some value within that range.*

3.4 Study 4 -- The Effect on Tracking Performance of the Operator's Criterion of Acceptable Performance

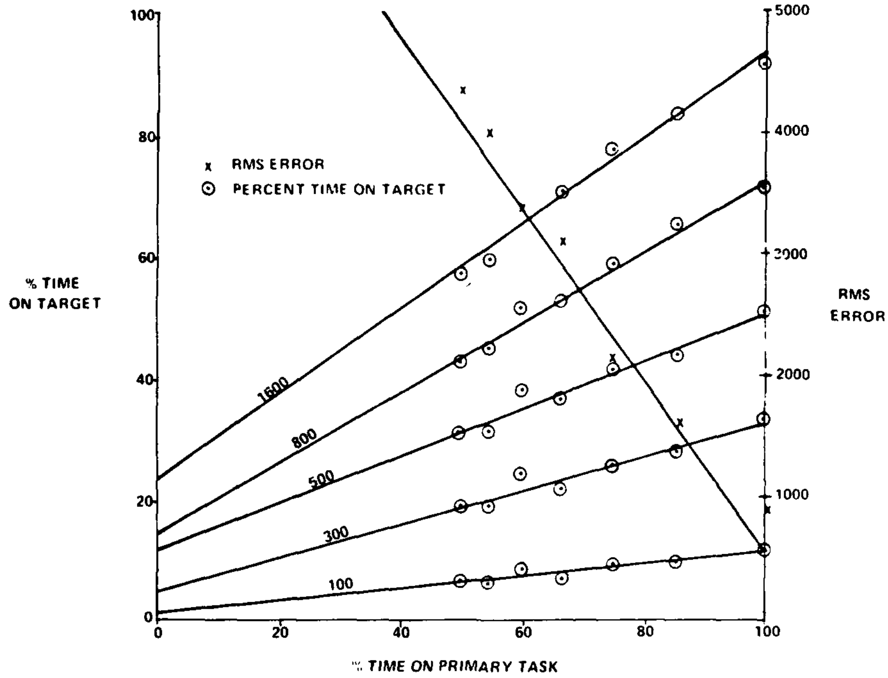
The effort that an operator will expend in performing a task is clearly dependent on what he considers acceptable performance to be. If the task is such that the operator has difficulty achieving acceptable performance, then we would expect him to devote all of his available effort to the task or else reassess his objectives. If acceptable performance can be achieved without devoting full effort to the task, then some of his reserve capacity might be available for another task. Therefore, if we were to give an operator a series of tracking tasks, each with a narrower criterion of acceptable performance, performance should improve until the criterion became smaller than the attainable performance. When too narrow a criterion was used, performance might degrade simply because the narrow criterion kept the operator from tracking the lower frequency. Such experiments would be virtually impossible to perform with a real human operator, since there is no reliable way to determine or control the internal criterion the operator is using. With HOS, however, the criterion of acceptable performance can be directly manipulated and its effects on tracking performance observed.

Study 4 examines the effects of varying the operator's internal criterion of successful performance for two different interruption cycles. The operator's criterion was varied from 0 to 800 in 200 unit increments. The two interruption cycles were six seconds of tracking

* The expected PTOT values are 19.0 percent for a criterion of 1600, 9.5 percent for a criterion of 800, 6.0 percent for a criterion of 500, 3.6 percent for a criterion of 300, and 1.2 percent for a criterion of 100.

TABLE 5. RESULTS OF STUDY 3

TRACKING		TASK ALTERNATION PERIOD'S DURATION				SUMMARY OF ANALYSIS OF CRITERION MEASURES							
							PERCENT TIME-ON-TARGET; SIZE:						
SIGNAL NO.	TOTAL % OF TIME	TOTAL	TRACK	DELAY	INTERNAL LIMITS	RMS ERROR	1600	800	500	300	100	CONTROL ADJUSTS
3	100.0	6.0	6.0	.0	800	910	91.7	71.5	50.4	32.8	11.3	625
3	85.7	7.0	6.0	1.0	800	1582	83.2	64.7	43.8	27.6	9.5	402
3	75.0	8.0	6.0	2.0	800	2109	77.3	58.3	40.9	25.4	9.0	362
3	66.7	9.0	6.0	3.0	800	3083	70.4	52.1	36.2	21.5	6.9	321
3	60.0	10.0	6.0	4.0	800	3326	66.8	51.5	37.9	24.0	8.2	309
3	54.5	11.0	6.0	5.0	800	3968	59.2	44.8	30.9	18.4	6.0	243
3	50.0	12.0	6.0	6.0	800	4318	56.9	42.4	30.5	18.5	6.2	220



CHANCE PERFORMANCE LEVELS

Criterion	TIME ON TARGET
1600	19.05
800	9.52
500	5.95
300	3.57
100	1.19

FIGURE 7. PERCENT TIME ON PRIMARY TASK VS. PERCENT TIME ON TARGET AND RMS ERROR

followed by a two second interruption and six seconds of tracking followed by a four second interruption. Signal 3 from Study 1 was again used for all trials.

The results of this study are shown in Table 6. Every decrease in the criterion produces an increase in the number of control adjustments but the performance measures remain essentially the same. Neither RMSE nor PTOT for the two largest target sizes show any systematic changes. However, PTOT for the three smallest target sizes (i.e., 500, 300, and 100 units) do show an improvement in tracking performance with each narrowing of the criterion and interruption cycle time.

The findings of this study are interesting in that they indicate how the apparent workload imposed by a tracking task can be changed when the operator adjusts his internal criterion of acceptable performance. When there are no secondary task interruptions, the operator may achieve acceptable performance with a relatively loose criterion tolerance. When the tracking task is periodically interrupted by the demands of other tasks, however, the operator may have to narrow his criterion in order to maintain performance. Of course, if it was the operator's responsibility to schedule the interruption sequence, he might relax his criterion of acceptable performance in order to minimize adjustment time. He could then work on the other tasks, thereby keeping each tracking interruption brief. Therefore, while it is apparent that the operator's workload increases whenever his criterion is narrowed (other things being equal), these studies indicate that measures like RMSE and PTOT might not show any change in performance.

TABLE 6. RESULTS OF STUDY 4

TRACKING		TASK ALTERNATION PERIOD'S DURATION				SUMMARY OF ANALYSIS OF CRITERION MEASURES						
							PERCENT TIME-ON-TARGET; SIZE:					
SIGNAL NO.	TOTAL % OF TIME	TOTAL	TRACK	DELAY	INTERNAL LIMITS	RMS ERROR	1600	800	500	300	100	CONTROL ADJUSTS
3	75.0	8.0	6.0	2.0	800	2109	77.3	58.3	40.9	25.4	9.0	362
3	75.0	8.0	6.0	2.0	600	2041	78.6	62.3	45.6	29.8	10.9	442
3	75.0	8.0	6.0	2.0	400	2123	78.3	60.2	44.1	30.0	12.1	511
3	75.0	8.0	6.0	2.0	200	1999	79.7	61.4	47.9	32.0	11.0	591
3	75.0	8.0	6.0	2.0	0	1923	78.6	62.1	49.6	35.9	12.9	696
3	60.0	10.0	6.0	4.0	800	3357	66.5	51.2	37.7	23.9	8.1	309
3	60.0	10.0	6.0	4.0	600	3224	64.2	48.0	35.2	22.4	7.5	343
3	60.0	10.0	6.0	4.0	400	3228	66.5	51.4	40.0	25.9	9.2	383
3	60.0	10.0	6.0	4.0	200	3'11	66.5	50.3	39.7	27.9	10.3	446
3	60.0	10.0	6.0	4.0	0	3393	67.4	52.1	42.0	30.9	11.7	569

4. CONCLUSION

The parametric studies in Section 2 and the workload studies in Section 3 demonstrate that the HOS tracking model exhibits the types of performance features that would be expected of a human operator. Some of its features were unexpected but no totally implausible findings were obtained. The instability of the learning process for some parameter values and for some signal characteristics is disturbing and deserves further investigation. The results obtained in these studies deserve to be replicated with real human operators in order both to validate the model and to corroborate our findings. Whether or not such research indicated that modifications to the tracking model were necessary, the potential usefulness of HOS in the assessment of the workload characteristics of crew station designs has been clearly demonstrated.

REFERENCES

Allport, D.A., Phenomenal simultaneity and the perceptual moment hypothesis. British Journal of Psychology, 1968, 59, pp. 395-406.

Analytics, Inc., The Human Operator Simulator. Volume II: HOS User's Guide, TR 1181-A, Analytics, Inc., Willow Grove, Pennsylvania, October 1975.

Baron, J., The threshold of successiveness. Perception and Psychophysics, 1971, 10 (4A), pp. 201-207.

Bush, R.R. and Mosteller, Fl, Stochastic Models for Learning, Wiley (New York), 1955.

Crossman, E.R.F.W., The information-capacity of the human motor-system in pursuit tracking. The Quarterly Journal of Experimental Psychology, 1960, 12, pp. 1-16.

Fitts, P.M. and Posner, M.I., Human Performance, Brooks/Cole Publishing Co. (Belmont, California), 1967.

Hick, W.E., The discontinuous functioning of the human operator in pursuit tasks. The Quarterly Journal of Experimental Psychology, 1948, 1, pp. 36-51.

Kristofferson, A.B., Attention and psychophysical time. Acta Psychologica, 1967, 27, pp. 93-100.

Linton, P., Jahns, D.W., and Chatelier, Pa., Operator workload assessment model: An evaluation of a VF/VA-V/STOL system. Presented at NATO/AGARD Aerospace Medical Panel Specialists Meeting, Koln, FRG, 18-22 April 1977.

Murphy, J.V. and Gurman, B.S., The integrated cockpit procedure for identifying control and display requirements of aircraft in advanced time periods. In: Guidance and Control Displays. NATO/AGARD Conference Proceedings CP-96, 1972.

Norman, M.F., On the linear model with two absorbing barriers. Journal of Mathematical Psychology, 1968, 5 (2), pp. 225-241.

North, J.D., Lomnicki, Z.A., and Zaremba, S.K., The design and interpretation of human control experiments. Ergonomics, 1958, 1, pp. 314-327.

Siegel, A.I. and Wolf, J.J., A model for digital simulation of two-operator man-machine systems. Ergonomics, 1962, 5, pp. 557-572.

Strieb, M.I., Glenn, F.A., Fisher, C., and Fitts, L., The Human Operator Simulator. Volume VII: LAMPS Air Tactical Officer Simulation, TR 1200, Analytics, Inc., Willow Grove, Pennsylvania, November 1976.

Stroud, J.M., The fine structure of psychological time. In: H. Quastler (ed.), Information Theory in Psychology, Free Press (New York), 1955.

APPENDIX I
HOPROC CODE AND HOS INPUTS FOR PARAMETRIC STUDIES

APPENDIX I

HOPROC CODE AND HOS INPUTS FOR PARAMETRIC STUDIES

This appendix contains a listing of the HOPROC code and HOS inputs used in the parametric studies presented in Section 2. A full discussion of HOPROC and HOS data card formats is presented in Volume II of this series, the HOS Users' Guide.

```
SYSTEM                HOS TEST CASE - GENERAL TRACKING MODEL
SETTING SECTION

OSTATE SECTION

ARGUMENT SECTION
       XDISP
       XCNTL

DISPLAY SECTION
       TRACK SCALE FEET
       SIGNAL_SCALE FEET
       SECONDARY

CONTROL SECTION
       START MOMENTARY
       SINE-WAVE-GENERATOR MOMENTARY
       KNOB

SYMBOL SECTION
```

OPERATOR FUNCTIONS

```
      GO TO 10000
9000  CONTINUE
      G = FST(¢NEW GAIN>)
      IF (G.GT.0.) GO TO 2
      RTIME = STIME + 10.
      IF (NG.LE.1) GO TO 1
      AVG = GSUM / NG
      SD = 0.
      VAR = (GSQ - NG*AVG*AVG) / (NG - 1.)
      IF (VAR.LE.0.) GO TO 1
      SD = SQRT(VAR)
1     CONTINUE
      WRITE (6,8000) AVG,SD,NG
8000  FORMAT (6X,*FOR COMPLETED PHASE MEAN GAIN =*,F10.2,4X,*SD =*,
     +     F10.2,4X,*OBSERVATIONS =*,I5)
      SDG = 0.
      IF (NG.LE.1) GO TO 3
      VAR = (NG-1.)*VAR/NG
      IF (VAR.LE.0.) GO TO 3
      SDG = SQRT(VAR)
      AVT = TSUM/NG
      SDT = 0.
      VAR = TSQ/NG - AVT*AVT
      IF (VAR.LE.0.) GO TO 3
      SDT = SQRT(VAR)
      AVGT = TGSUM/NG
      COR = (AVGT-AVG*AVT)/(SDG*SDT)
      REG = (AVGT-AVG*AVT)/VAR
3     CONTINUE
      WRITE (6,8003) COR,REG,AVT,SDG,SDT
8003  FORMAT (10X,*COR.COEFF. =*,F8.5,5X,*REG.COEFF. =*,F10.4,5X,
     +     *AVT =*,F10.2,5X,*SDG =*,F10.2,5X,*SDT =*,F10.2)
      NG = 0
      GSUM = GSQ = 0.
      TSUM = TSQ = TGSUM = 0.
      G = FST(9) = 200.
2     CONTINUE
      XT = TFREE(4) - STIME
      IF (XT.GT.0.) CALL BANKER(XT)
      PE = CE
      ID = ¢TRACK>
      KD = ¢SIGNAL>
      CE = FST(ID) - FST(KD)
```

```
      AVGERR = .9 * AVGERR + .1 * ABS(CE)
      IF (AVGERR.LT.SMALL) AVGERR=SMALL
      THETA = ATHET * ABS(CE) / AVGERR
      IF (THETA.GT..9) THETA=.9
      IF (THETA.LT.0.) THETA=0.
      XNG = 1.0
      IF (PE.EQ.0..AND.CE.EQ.0.) GO TO 8
      IF ((CE.GE.PE.AND.PE.GE.0.).OR.(CE.LE.PE.AND.PE.LE.0.)) GO TO 6
      XNG=1.0-CE/PE
      IF (XNG.GT.2.) XNG = 2.
      IF (XNG.LT..5) XNG = .5
      GO TO 8
6     XRN = RANF(DUMMY)
      IF (XRN.GT..5) XNG=ADELT
      IF (XRN.LT..5) XNG=SDELT
8     G = THETA*G*XNG + (1.0-THETA)*G
      IF (STIME.LT.RTIME) GO TO 4
      GSUM = GSUM + G
      GSQ = GSQ + G*G
      TSUM = TSUM + STIME
      TSQ = TSQ + STIME*STIME
      TGSUM = TGSUM + G*STIME
      NG = NG + 1
4     CONTINUE
      "NEW GAIN" = G

      GN = EST(¢NEW GAIN>)
      IF (GN.LT.SMALL) GN=SMALL
      "KNOB CHANGE" = (DESIRE(¢TRACK>) - EST(¢TRACK>))/GN

      READ (5,500) THETA,DELTA,EXRATE,DELAY
500   FORMAT (4F8.3)
      WRITE (6,501) THETA,DELTA,EXRATE,DELAY
501   FORMAT (//* PARAMETERS FOR THIS RUN ARE AS FOLLOWS:*//* THETA =*,
     +    F8.3/* DELTA =*F8.3/* EXRATE =*,F8.3,/* DELAY =*,F8.3)
      ADELT = 1.0 + DELTA
      SDELT = 1.0 / ADELT
      ATHET = THETA
      AVGERR = 0.
      "PARAMETERS" = THETA

      ID = ¢SIGNAL>
      ET = ESTIME(ID) - PESTIM(ID)
      IF (ET.LT.SMALL) ET=SMALL
      KD = ¢EXTIME>
      RTIME = EST(KD)
      XX = EST(ID)+(EST(ID)-PEST(ID))*(STIME+RTIME-ESTIME(ID))/ET
      "EXVAL" = XX
```

```
      IF (EST(12).GT.0.) GO TO 12
      PTIME = STIME + 10.
      IF (NX.LE.1) GO TO 11
      AVG = XSUM / NX
      SD = 0.
      VAR = (XSQ - NX*AVG*AVG) / (NX - 1.)
      IF (VAR.LE.0.) GO TO 11
      SD = SQRT(VAR)
11    CONTINUE
      WRITE (6,8001) AVG,SD,NX
8001  FORMAT (6X,*FOR COMPLETED PHASE MEAN EXTIME =*,F10.4,4X,*SD =*,
     +      F10.4,4X,*OBSERVATIONS =*,I5)
      SDX = 0.
      IF (NX.LE.1) GO TO 13
      VAR = (NX-1.)*VAR/NX
      IF (VAR.LE.0.) GO TO 13
      SDX = SQRT(VAR)
      AVT = TTSUM/NX
      SDT = 0.
      VAR = TTSQ/NX - AVT*AVT
      IF (VAR.LE.0.) GO TO 13
      SDT = SQRT(VAR)
      AVXT = TXSUM/NX
      COR = (AVXT-AVG*AVT)/(SDX*SDT)
      REG = (AVXT-AVG*AVT)/VAR
13    CONTINUE
      WRITE (6,8003) COR,REG,AVT,SDX,SDT
      NX = 0
      XSUM = XSQ = 0.
      TTSUM = TTSQ = TXSUM = 0.
      EST(12) = .1
12    CONTINUE
      KD = ¢EXTIME>
      ID = ¢TRACK>
      JD = ¢SIGNAL>
      PS = CS
      CS = EST(JD)
      CERR = EST(ID) - EST(JD)
      EXT = EST(KD)
      IF (AVGERR.LT.SMALL) AVGERR = SMALL
      XLR = EXRATE*CERR/AVGERR
      IF (XLR.GT..2) XLR=.2
      IF (XLR.LT.-.2) XLR=-.2
      IF (CS.GT.PS) XLR=-XLR
      X = (1.+XLR)*EXT
      IF (STIME.LT.PTIME) GO TO 14
      XSUM = XSUM + X
      XSQ = XSQ + X*X
      TTSUM = TTSUM + STIME
      TTSQ = TTSQ + STIME*STIME
      TXSUM = TXSUM + X*STIME
      NX = NX + 1
14    CONTINUE
      "EXTIME" = X
```

```
CALL BANKER(DELAY)
INDPRO = 1
"SECONDARY DELAY" = DELAY

II = ¢TACCUR>
X = EST(II)
"TACCUR" = X

HARDWARE FUNCTIONS
      GO TO 10000
9000  CONTINUE
      I = ¢KNOB>
      IF(TEREF(IWANT)-STIME.EQ.0)RETURN
      "RATE1"=(DESIRE(I)-EST(I))/(TEREF(IWANT)-STIME)

      IM = MODEL(¢KNOB>)
      GAIN = PARA(IM,9)
      "RATE2" = GAIN*"RATE1"

      I=¢XCNTL>
      CALL REF(I)
      "UPDATEC"=RATE(I)*(STIME-TIME(I))

      I=¢XCNTL>
      CALL REF(I)
      J=¢XDISP>
      CALL REF(J)
      "UPDATED"=RATE(J)*(STIME-TIME(I))

      READ(5,100) AMP,FREQ,OFFSET
100   FORMAT (3(F8.2,1X))
      WRITE (6,101) AMP,FREQ,OFFSET
101   FORMAT (* AMP =*,F8.2/* FREQ =*,F8.2/* OFFSET =*,F8.2//)
      "SINE-WAVE-FACTORS"=1

      KK = STLAST/60.
      LL = STIME/60.
      IF (KK.EQ.LL) GO TO 200
      LL = LL + 1
      WRITE (6,2000) LL,EST(12),EST(9)
2000  FORMAT (1X,*BEGINNING PHASE *,I1,4X,*EXTIME =*,F10.4,6X,
     +      *GAIN =*,F10.2)
      FREQ = FREQ*1.41421356
      WRITE (6,2001) FREQ
2001  FORMAT (1X,*NEW SIGNAL FREQUENCY =*,F12.8)
      EST(9) = EST(12) = -1.
200   CONTINUE
      STLAST = STIME
      "NEW SIGNAL" = AMP*SIN(STIME*FREQ) + OFFSET
```

```
HARDWARE PROCEDURES
        DEFINE PROCEDURE TO SIMULATE START.
END:    DETERMINE SINE-WAVE-FACTORS.

        DEFINE PROCEDURE TO SIMULATE SINE-WAVE-GENERATOR.
MID:    DETERMINE NEW-SIGNAL.
        SET SIGNAL TO NEW-SIGNAL.

        DEFINE PROCEDURE TO SIMULATE KNOB.
START:   DETERMINE RATE1.
         SET RATE OF KNOB TO RATE1.
         DETERMINE RATE2.
         SET RATE OF TRACK TO RATE2.
MIDEND: COMPUTE-UPDATE-DC USING TRACK, KNOB.
END:     SET RATE OF KNOB TO 0.

        DEFINE COMPUTE-UPDATE-DC USING XDISP,XCNTL.
        DETERMINE UPDATEC.
        INCREASE XCNTL BY UPDATEC.
        DETERMINE UPDATED.
        INCREASE XDISP BY UPDATED.
```

```
OPERATOR PROCEDURES
      DEFINE MISSION.
          COMPUTE PARAMETERS.
          DEPRESS START.
          DEPRESS SINE-WAVE-GENERATOR.
          SET LIMITS OF TRACK TO TACCUR.
          MONITOR TRACK.
          MONITOR SECONDARY.
          IF TIME OF SIMULATION IS LESS THAN 10000 SECONDS THEN WAIT.

      DEFINE PROCEDURE TO ADJUST TRACK.
          READ SIGNAL.
          COMPUTE EXTIME.
          COMPUTE EXVAL.
          SET TRACK TO RESULT.
          READ TRACK.
          IF TRACK IS WITHIN LIMITS THEN GO TO 1 NOW.
          COMPUTE NEW-GAIN.
          COMPUTE KNOB-CHANGE.
          TURN KNOB BY RESULT.
1:        COMPUTE TACCUR.

      DEFINE PROCEDURE TO ADJUST SECONDARY.
          COMPUTE SECONDARY-DELAY.
```

```
SYSTEM                  PURSUIT TRACKING MODEL
DISPLAY SECTION
TRACK                   1 0.1 1 12 12 12 10000.
SIGNAL                  4 0 1 1 12 12 12 0
SECONDARY               1 0 1 1 0 0 0 0
CONTROL SECTION
START                   2 0 1 1 -15 12 0 0
SINE WAVE GENERATOR 5 0 1 1 -12 12 0 0
KNOB                    3 0 1 1 -6 12 0 0
OPERATOR FUNCTIONS
EXTIME                  2 0 1 1 0 .1
EXVAL                   2 0 1 1 0 0
KNOB CHANGE             2 0 1 1 0 0
NEW GAIN                2 0 1 1 0 200.
PARAMETERS              2 0 1 1 0 1.0
SECONDARY DELAY         2 0 1 1 0 0
TACCUP                  2 0 1 1 0 0
PROCEDURE SECTION
MISSION                 1.0
 TRACK                  1.0
 SECONDARY              1.0
MODEL SPECIFICATIONS
MODEL 1                 4 1 500 .015 2.0
MODEL 2                 7 3 0 .01 1. 0 0 .01
MODEL 3                 9 2  1 0  2. 0 0   300 10 200
MODEL 4                 4 1 500 .015 5.0
MODEL 5                 7 4 5 .01 1.0 0 0  999.
END OF MODEL SPEX
HUMAN OPERATOR SPEX
EYES                    12 0 0 .01 33 48
HANDS                   6 6 -18 -6 6 -18
SHOULDERS               6 0 -6 -6 0 -6
END OF HUMAN SPEX
INTERRUPTION DATA       0.0 6.0
RUN PARAMETERS          .07 .9 .01 2. .2 1. .01 5. 5. 10. 10. 10. 3. 3.  .01
PLOT ASTERISKS
PRINT MESSAGES
TIMED ENDPOINT          181
$
.0         .0        .05
4000.0     2.513272 10000.0
#
```

APPENDIX II
HOPROC CODE AND HOS INPUTS FOR WORKLOAD STUDIES

APPENDIX II
HOPROC CODE AND HOS INPUTS FOR WORKLOAD STUDIES

This appendix contains a full listing of the HOPROC code and HOS inputs used in the workload studies described in Section 3 of this report. A full discussion of HOPROC and HOS data card fromats is presented in Volume II of this series, the HOS Users' Guide.

```
SYSTEM                 HOS TEST CASE - GENERAL TRACKING MODEL
SETTING SECTION

QSTATE SECTION

ARGUMENT SECTION
       XDISP
       XCNTL

DISPLAY SECTION
       TRACK SCALE FEET
       SIGNAL SCALE FEET
       SECONDARY

CONTROL SECTION
       START MOMENTARY
       SINE-WAVE-GENERATOR MOMENTARY
       KNOB

SYMBOL SECTION
```

```
OPERATOR FUNCTIONS
      GO TO 10000
9000  CONTINUE
      G = EST(<NEW GAIN>)
      XT = TFREE(4) - STIME
      IF (XT.GT.0.) CALL BANKER(XT)
      PE = CE
      ID = <TRACK>
      KD = <SIGNAL>
      CE = EST(ID) - EST(KD)
      IF (IDELAY.EQ.1) GO TO 9
      AVGERR = .9 * AVGERR + .1 * ABS(CE)
      IF (AVGERR.LT.SMALL) AVGERR=SMALL
      THETA = ATHET * ABS(CE) / AVGERR
      IF (THETA.GT..9) THETA=.9
      IF (THETA.LT.0.) THETA=0.
      XNG = 1.0
      IF (PE.EQ.0..AND.CE.EQ.0.) GO TO 8
      IF ((CE.GE.PE.AND.PE.GE.0.).OR.(CE.LE.PE.AND.PE.LE.0.)) GO TO 6
      XNG=1.0-CE/PE
      IF (XNG.GT.2.) XNG = 2.
      IF (XNG.LT..5) XNG = .5
      GO TO 8
6     XRN = RANF(DUMMY)
      IF (XRN.GT..5) XNG=ADELT
      IF (XRN.LT..5) XNG=SDELT
8     G = THETA*G*XNG + (1.0-THETA)*G
9     CONTINUE
      IF (IDELAY.GT.0) IDELAY = IDELAY - 1
      "NEW GAIN" = G

      GN = EST(<NEW GAIN>)
      IF (GN.LT.SMALL) GN=SMALL
      "KNOB CHANGE" = (DESIRE(<TRACK>) - EST(<TRACK>))/GN

      READ (5,500) THETA,DELTA,EXRATE,DELAY
500   FORMAT (4F8.3)
      WRITE (6,501) THETA,DELTA,EXRATE,DELAY
501   FORMAT (//* PARAMETERS FOR THIS RUN ARE AS FOLLOWS:*//* THETA =*,
     +    F8.3/* DELTA =*F8.3/* EXRATE =*,F8.3,/* DELAY =*,F8.3)
      ADELT = 1.0 + DELTA
      SDELT = 1.0 / ADELT
      ATHET = THETA
      AVGERR = 0.
      IDELAY = 0
      "PARAMETERS" = THETA
```

```
ID = <SIGNAL>
ET = ESTIME(ID) - PESTIM(ID)
IF (ET.LT.SMALL) ET=SMALL
KD = <EXTIME>
PTIME = EST(KD)
XX = EST(ID)+(EST(ID)-PEST(ID))*(STIME+PTIME-ESTIME(ID))/ET
"EXVAL" = XX

KD = <EXTIME>
ID = <TRACK>
JD = <SIGNAL>
PS = CS
CS = EST(JD)
CERR = EST(ID) - EST(JD)
EXT = EST(KD)
IF (AVGERR.LT.SMALL) AVGERR = SMALL
XLR = EXRATE*CERR/AVGERR
IF (XLR.GT..2) XLR=.2
IF (XLR.LT.-.2) XLR=-.2
IF (CS.GT.PS) XLR=-XLR
IF (IDELAY.EQ.1) XLR = 0.
"EXTIME" = (1.+XLR)*EXT

CALL BANKER(DELAY)
IDELAY = 1
INDPRO = 1
"SECONDARY DELAY" = DELAY

II = <TACCUR>
X = EST(II)
"TACCUR" = X
```

```
HARDWARE FUNCTIONS
      GO TO 10000
9000  CONTINUE
      I = ¢KNOB>
      IF(TFREE(IWANT)-STIME.EQ.0)RETURN
      "RATE1"=(DESIRE(I)-EST(I))/(TFREE(IWANT)-STIME)

      IM = MODEL(¢KNOB>)
      GAIN = PARA(IM,9)
      "RATE2" = GAIN*"RATE1"

      I=¢XCNTL>
      CALL REF(I)
      "UPDATEC"=RATE(I)*(STIME-TIME(I))

      I=¢XCNTL>
      CALL REF(I)
      J=¢XDISP>
      CALL REF(J)
      "UPDATED"=RATE(J)*(STIME-TIME(I))

      READ(5,100) AMP,FREQ,OFFSET
100   FORMAT (3(F8.2,1X))
      WRITE (6,101) AMP,FREQ,OFFSET
101   FORMAT (* AMP =*,F8.2/* FREQ =*,F8.2/* OFFSET =*,F8.2//)
      "SINE-WAVE-FACTORS"=1

      XA = SIN(STIME*FREQ)
      XB = .7*SIN(10.*STIME*FREQ/7.)
      XC = .4*SIN(2.5*STIME*FREQ)
      "NEW SIGNAL" = AMP*(XA+XB+XC) + OFFSET
```

```
HARDWARE PROCEDURES
       DEFINE PROCEDURE TO SIMULATE START.
END:   DETERMINE SINE-WAVE-FACTORS.

       DEFINE PROCEDURE TO SIMULATE SINE-WAVE-GENERATOR.
MID:   DETERMINE NEW-SIGNAL.
       SET SIGNAL TO NEW-SIGNAL.

       DEFINE PROCEDURE TO SIMULATE KNOB.
START:   DETERMINE RATE1.
         SET RATE OF KNOB TO RATE1.
         DETERMINE RATE2.
         SET RATE OF TRACK TO RATE2.
MIDEND: COMPUTE-UPDATE-DC USING TRACK, KNOB.
END:     SET RATE OF KNOB TO 0.

       DEFINE COMPUTE-UPDATE-DC USING XDISP,XCNTL.
       DETERMINE UPDATEC.
       INCREASE XCNTL BY UPDATEC.
       DETERMINE UPDATED.
       INCREASE XDISP BY UPDATED.
```

```
OPERATOR PROCEDURES
      DEFINE MISSION.
          COMPUTE PARAMETERS.
          DEPRESS START.
          DEPRESS SINE-WAVE-GENERATOR.
          SET LIMITS OF TRACK TO TACCUR.
          MONITOR TRACK.
          MONITOR SECONDARY.
          IF TIME OF SIMULATION IS LESS THAN 10000 SECONDS THEN WAIT.

      DEFINE PROCEDURE TO ADJUST TRACK.
          READ SIGNAL.
          COMPUTE EXTIME.
          COMPUTE EXVAL.
          SET TRACK TO RESULT.
          READ TRACK.
          IF TRACK IS WITHIN LIMITS THEN GO TO 1 NOW.
          COMPUTE NEW-GAIN.
          COMPUTE KNOB-CHANGE.
          TURN KNOB BY RESULT.
1:        COMPUTE TACCUR.

      DEFINE PROCEDURE TO ADJUST SECONDARY.
          COMPUTE SECONDARY-DELAY.
```

```
SYSTEM                 PURSUIT TRACKING MODEL
DISPLAY SECTION
TRACK                  1 0 1 1 12 12 12 10000.
SIGNAL                 4 0 1 1 12 12 12 0
SECONDARY              1 0 1 1 0 0 0 0
CONTROL SECTION
START                  2 0 1 1 -15 12 0 0
SINE WAVE GENERATOR 5 0 1 1 -12 12 0 0
KNOB                   3 0 1 1 -6 12 0 0
OPERATOR FUNCTIONS
EXTIME                 2 0 1 1 0 .1
EXVAL                  2 0 1 1 0 0
KNOB CHANGE            2 0 1 1 0 0
NEW GAIN               2 0 1 1 0 200.
PARAMETERS             2 0 1 1 0 1.0
SECONDARY DELAY        2 0 1 1 0 0
TACCUS                 2 0 1 1 0 0
PROCEDURE SECTION
MISSION                1.0
 TRACK                 1.0
 SECONDARY             1.0
MODEL SPECIFICATIONS
MODEL 1                4 1 500 .015 2.0
MODEL 2                7 3 0 .01 1. 0 0 .01
MODEL 3                9 2  1 0  2. 0 0   300 10 200
MODEL 4                4 1 500 .015 5.0
MODEL 5                7 4 5 .01 1.0 0 0  999.
END OF MODEL SPEX
HUMAN OPERATOR SPEX
EYES                   12 0 0 .01 33 42
HANDS                  6 6 -18 -6 6 -18
SHOULDERS              6 0 -6 -6 0 -6
END OF HUMAN SPEX
INTERRUPTION DATA      0.0 6.0
RUN PARAMETERS         .07 .9 .01 2. .2 1. .01 5. 5. 10. 10. 10. 3. 3.  .0
PLOT ASTERISKS
PRINT MESSAGES
TIMED ENDPOINT         181
$
.0        .0        .05
4000.0    2.513272 10000.0
#
```

APPENDIX III
CHANGES TO HOS FOR TRACKING AND WORKLOAD STUDIES

APPENDIX III
CHANGES TO HOS FOR TRACKING AND WORKLOAD STUDIES

The interrupted tracking simulations described in Section 3 required several minor modifications to HOS. These changes replaced the normal operator-controlled sequencing of procedures by a time-controlled sequence dependent upon an input alternation period. The modified version of HOS was also used for the parametric studies reported in Section 2 for which the duration of each interrupt was set to zero. The changes made to the standard version of HOS were:

(1) Four new variables -- TPRIM, TBETW, DURINT, and INDPRO were added to COMMON JUNK. TPRIM is the time when each execution of the primary procedure begins. TBETW is the time between interrupts (i.e., the time spent on the primary task for each alternation period). DURINT is the duration of each interrupt. INDPRO is the indicator of which procedure should be executed at any time (INDPRO = 1 for the primary procedure and INDPRO = 2 for the secondary procedure).

(2) Four cards

```
    IF (INDPRO.EQ.2.OR.STIME.LT.(TPRIM+TBETW)) GO TO 1025

    INDPRO = 2

    GO TO 1040

1025 CONTINUE
```

were inserted immediately after statement label 1020 in the HOS main routine.

(3) Three cards

```
IF(IP.NE.28) GO TO 280

TPRIM = STIME

INDPRO = 1
```

were inserted immediately prior to statement label 280 in subroutime ENDPROC.

(4) One card

```
IF(MARGIN(1).EQ.10HINTERRUPTI) GO TO 820
```

was inserted in the list of IF statements between statement lables 301 and 310 in the subroutine INPROC.

(5) Six cards

```
820 CONTINUE
    CALL NEWORD
    DURINT = DIGITS
    CALL NEWORD
    TBETW = DIGITS
    GO TO 290
```

were inserted immediately prior to the END card in the subroutine INPROC.

(6) One card

```
IF(IC.NE.1) IC=INDPRO+1
```

These changes require one additional input data card that is not required by the standard version of HOS. That card has the words INTERRUPTION DATA starting in column 1 and two numbers starting in column 21. The first number specifies the length of each interruption and the second number specifies the length of each primary task execution.

APPENDIX IV
DESCRIPTION AND LIST OF EVAL

APPENDIX IV
DESCRIPTION AND LIST OF EVAL

The analysis program EVAL computes a variety of tracking performance statistics from data appearing on one of the output files created by HOS. This file, originally created for use by the HODAC graph-generating subroutine TGRAPH, is a series of two-word (60 bits per word) micro-reports that describe simulation events. The low-order twelve bits of the first word contains a message number that identifies the report. The only message numbers recognized by EVAL are:

NUMBER	MESSAGE
13	START TO COMPUTE ...
35	BEGIN MANIPULATING ... WITH LEFT HAND
36	COMPLETE MANIPULATING ... WITH LEFT HAND
62	COMPLETE COMPUTATION OF ...
80	ACTUAL VALUE OF ... CHANGED

Other messages appearing on the file are ignored by EVAL.

The next twelve bits of the first word contain the dictionary entry number of the device, function, or procedure referenced by the message. The highest order 30 bits contain the simulation time at which the message was produced. For messages 13, 35, 36, and 62, the second word contains the estimated value (EST) of the referenced item in its low order 50 bits and the integerized value of the hab strength of the item multiplied by 1,000 in its high order 10 bits. For message number 80, the full 60 bits of the second word are used to represent the new actual value of the referenced item.

The program EVAL consists of four routines. The main routine, EVAL, processes all input data cards, monitors the operations of tape reading and data accumulation, and prints out summary statistics. The subroutine EDATA accumulates all RMSE and PTOT performance statistics according to directives from the main routine. The subroutine TREAD reads the next micro-report on the HOS output tape and transfers the message data to the calling routine via the formal arguments of the subroutine. The function SIGNAL determines the value of the signal being tracked at any given simulation time.

Input data specifying the parameters of the signal being tracked and the type of analysis to be performed are required by EVAL. Each input data card must contain an identifying label starting in column 1 and a single number starting in column 21. The recognized identifying labels are:

INTERVAL

DELAY

START

END

SIGNAL OFFSET

SIGNAL FREQ1

SIGNAL FREQ2

SIGNAL FREQ3

SIGNAL AMP1

SIGNAL AMP2

SIGNAL AMP3

LIMIT

TIMELINE

The INTERVAL card specifies the size of each data collection interval. The DELAY card indicates the length of time at the beginning of each primary task execution before the first data collection interval for that execution will begin. The START and END cards specify the simulation times over which the analysis is to be performed. The cards that specify signal characteristics are self-explanatory. Each LIMIT card (up to five are allowed) specifies a PTOT criterion tolerance to be used in the analysis. The TIMELINE card (which needs no number in its second field) indicates that a timeline analysis is to be printed while the tape is being processed. No timeline report is printed if the TIMELINE card is omitted.

The logic of the program is reasonably straightforward and will not be described in detail. Comment cards in each routine indicate the function performed by each section of code. In order to assist the programmer in interpreting and modifying the program, the following list of variable descriptions is offered. After the list of variable descriptions is a full list of the EVAL program followed by a sample program output.

VARIABLE NAME	DESCRIPTION
AEND	end time for analysis
AMP(I)	amplitudes of the three sine wave components of the signal
ASTART	start time for analysis
BTIM(I)	start time for I^{th} buffer interval during a movement
BUFCTI(I)	CTINT for the I^{th} buffer interval
BUFEAR(I)	error at return to primary for I^{th} buffer interval
BUFINIT(I)	indication of whether (1) or not (0) interval data accumulators are to be initialized after processing data for I^{th} buffer interval
BUFTSR(I)	time since return to primary task for I^{th} buffer interval
CAERRT	cumulative absolute error at initiations of primary procedure
CAESEC	cumulative absolute error at initiations of secondary procedure
CLEVEL	value of "track" when at a constant value between movements
CTINT	time when current data collection interval began
CUECI	cumulative squared error for current interval
CUEPRI	cumulative squared error for primary procedure
CUESEC	cumulative squared error for secondary procedure
ECI	cumulative error for current interval
ECUM(I)	cumulative error for primary procedure if I = 1 and for secondary procedure if I = 2

VARIABLE NAME	DESCRIPTION
ECUM(I,J)	cumulative error for J^{th} interval after start of primary procedure if I = 1 or of secondary procedure if I = 2
ERRET	actual error when current execution of primary procedure began
ETIM(I)	end time for I^{th} buffer interval
IINT(I)	NINT for I^{th} buffer interval
IMOVE	1 if a movement is in progress, 2 if a movement has been completed but data has not been stored because "actual value changed" message has not yet been encountered, and 0 otherwise
INTEND(I)	1 if the I^{th} buffer interval marks the end of a timed interval and 0 otherwise
ITLIM(I)	integerized value of the I^{th} criterion tolerance (TLIM(I)) for PTOT determinations
MARGIN(I)	the I^{th} (first or second) word in the identifier field of an input data card
MOVEALL	counter of all control movements over time span of analysis
MOVECI	counter of movements for current interval
NAERRT	number of initiations of primary procedure
NAESEC	number of initiations of secondary procedure
NBUF	number of items in movement buffer
NCI	number of error observations for current interval
NCPRI	number of observations of squared error for primary procedure
NCSEC	number of observations of squared error for secondary procedure

VARIABLE NAME	DESCRIPTION
NINT	Current interval number after beginning of executing procedure (maximum of 20) if interval data is being collected -- value is positive for primary procedure and negative for secondary procedure -- value of 0 indicates primary procedure is executing but no data should be collected and value of -30 indicates that secondary procedure is executing but no data should be collected
NLIMIT	the number of PTOT criterion limits specified in the input data
NOBS(I)	number of observations of RMSE measures for I^{th} interval after start of primary procedure
NSEC(I)	number of observations of RMSE measures for I^{th} interval after start of secondary procedure
NTLINE	number of lines on current page of time-line output
OFFSET	signal offset
PCINT	time when current processing chunk of current data collection interval began
PRIMT	total time spent on primary procedure
SAERRT	sum of squares of absolute errors at initiations of primary procedure
SAESEC	sum of squares of absolute errors at initiations of secondary procedure
SCENI(I)	sum of RMSE measures for all observations of I^{th} interval after start of primary procedure
SECT	total time spent on secondary procedure
SESEC(I)	sum of RMSE measures for all observations of I^{th} interval after start of secondary procedure
SF(I)	frequencies of the three sine wave components of the signal

VARIABLE NAME	DESCRIPTION
SMOVEI(I)	sum of numbers of observed control movements over all observations of the I^{th} interval after the start of the primary procedure
SSCENI(I)	sum of squares of RMSE measures for all observations of the I^{th} interval after start of primary procedure
SSESEC(I)	sum of squares of RMSE measures for all observations of I^{th} interval after start of secondary procedure
SSMOVEI(I)	sum of squares of numbers of observed control movements over all observations of the I^{th} interval after the start of the primary procedure
SSTTNI(I,J)	sum of squares of the percent times on target with respect to criterion tolerance J for all observations of the I^{th} interval after the start of the primary procedure
SSTTSEC(I,J)	sum of squares of the percent times on target with respect to criterion tolerance J for all observations of the I^{th} interval after the start of the secondary procedure
STTNI(I,J)	sum of the observed percent times on target with respect to criterion tolerance J for all observations of the I^{th} interval after the start of the primary procedure
STTSEC(I,J)	sum of the observed percent times on target with respect to criterion tolerance J for all observations of the I^{th} interval after the start of the secondary procedure
TBMOVE	time when current movement began
TDEL	user-specified delay after beginning of primary task when first data collection interval begins
TEMOVE	time when most recent movement ended

VARIABLE NAME	DESCRIPTION
TINT	user-specified duration of each data collection interval
TLAST	time of the previous micro-report message
TLIM(I)	the user-specified value of the I^{th} criterion tolerance (in the order specified by the user) for PTOT determinations
TOTCI(I)	cumulative number of "on-target" observations with respect to the I^{th} criterion tolerance for current data collection interval (observations are regularly spaced at .01 second intervals)
TOTPRI(I)	cumulative number of "on target" observations with respect to the I^{th} criterion tolerance for all time on the primary procedure
TOTSEC(I)	cumulative number of "on-target" observations with respect to the I^{th} criterion tolerance for all time on the secondary procedure
TPRIM	0 if the primary procedure is not executing and otherwise is the time that the current execution began
TSECO	0 if the secondary procedure is not executing and otherwise is the time that the current execution began
T1, T2, V1, V2, NN, INTCUM (formal arguments of EDATA)	The track element moves along the straight line connecting the points (T1, V1) and (T2, V2) where V1 is the track value at time T1 and V2 is the track value at time T2. NN is the interval number relative to the start of the procedure for which data is to be accumulated. If INTCUM=1 data for the current interval should be stored and accumulator variables should be initiated, but not if INTCUM=0
ST, IM, ID, VAL (formal arguments of TREAD)	Data values obtained by TREAD from reading of next micro-report on HOS output tape. ST is simulation time of report. IM is the code number of the message (see text above for interpretation of code). ID is the dictionary pointer referenced by

VARIABLE NAME	DESCRIPTION
ST, IM, ID, VAL (continued)	the report. VAL is the value (estimated or actual) of the item referenced by the report
T (function of argument of SIGNAL)	the simulation time for which the signal value is to be returned

```
      PROGRAM EVAL(TAPE12,INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON /STORE/CUECI,TOTCI(5),NCI,CUEPRI,TOTPRI(5),NCPRI,CUESEC,
     +    TOTSEC(5),NCSEC,SCENI(20),SSCENI(20),STTNI(20,5),SSTTNI(20,5)
     +    NOBS(20),TLIM(5),NLIMIT,MOVECI,SMOVEI(20),SSMOVEI(20)
     +    ,SESEC(20),SSESEC(20),STTSEC(20,5),SSTTSEC(20,5),NSEC(20)
     +    ,ECUM(2),ECUMI(2,20),NIERR(2,20),ECI
      COMMON /SIG/SF(3),AMP(3),OFFSET
      DIMENSION MARGIN(2),BTIM(10),ETIM(10),INTEND(10),IINT(10),
     +    BUFTSR(10),BUFFAR(10),BUFCTI(10),PTOT(5),PTOTSD(5),ITLIM(5)
     +    ,BUFINIT(10)
      DATA TINT/1.0/,TDEL/0./,ASTART/0./,AEND/99999./,
     +    OFFSET/0./,SF/0.,0.,0./,NTLINE/60/,IEE/77777777770000000000B/

9000  FORMAT (2A10,F13.6)
9001  FORMAT (1X,2A10,F13.6)
9002  FORMAT (1X,*$$$ ERROR. SIGNAL DATA HAS NOT BEEN FULLY SPECIFIED*)
9003  FORMAT (1H1,29X,*TIMELINE DATA FOR PRIMARY TASK BY TIME INTERVAL*
     +    //8X,*INTERVAL*,6X,*TIME SINCE  ERROR AT*,4X,*NO.*,8X,*RMS  *
     +    5(4X,*PERCENT*)/4X,*START*,6X,*END*,6X,*RETURN*,5X,*RETURN*,
     +    4X,*MOVES*,6X,*ERROR*,4X,5(*TOT(*,I4,*)*,2X))
9004  FORMAT (/F8.2,2X,F8.2,4X,F7.2,3X,F9.2,3X,I5,4X,F8.2,1X,5(3X,F8.3)
9005  FORMAT (1X,*$$$ WARNING. OVERFLOW IN MOVEMENT BUFFER.*,4X,
     +    *ST=*,F8.2,3X,*IM=*,I3,3X,*ID=*,I3,3X,*VAL=*,F8.2)
9006  FORMAT (1X,*$$$ ERROR. UNDERFLOW IN MOVEMENT BUFFER.*,4X,
     +    *ST=*,F8.2,3X,*IM=*,I3,3X,*ID=*,I3,3X,*VAL=*,F8.2)
9007  FORMAT (1H1,28X,*CUMULATIVE DATA FOR *,A9,* TASK BY TIME *,
     +    *INTERVAL*//* INTERVAL  NUMBER OF  TIME SINCE*,4X,*NUMBER OF*
     +    8X,*RMS  *,5(6X,*PERCENT*)/2X,*NUMBER*,5X,*OBS.*,7X,
     +    *RETURN*,6X,*MOVEMENTS*,7X,*ERROR*,
     +    2X,5(4X,*TOT(*,I4,*)*))
9008  FORMAT (/4X,I2,6X,I5,6X,F7.2,6X,F8.2,6X,F8.2,5(5X,F8.3))
9009  FORMAT (1H1,41X,*CUMULATIVE DATA BY PROCEDURE*//17X,*TOTAL*,
     +    3X,*ABSOLUTE ERROR AT ENTRY*,5X,*NO.*,8X,*RMS  *,
     +    5(4X,*PERCENT*)/* PROCEDURE*,8X,*TIME*,8X,*MEAN*,6X,*S.D.*,
     +    8X,*MOVES*,6X,*ERROR  *,5(2X,*TOT(*,I4,1H)))
9010  FORMAT (/1X,A10,4X,F8.2,4X,F8.2,2X,F8.2,6X,I6,4X,F8.2,5(3X,F8.3))
9011  FORMAT (1X,*TOO MANY LIMIT VALUES - ONLY 5 ARE ALLOWED.*)
9012  FORMAT (31X,2(4X,1H(,F8.2,1H)),5(4X,1H(,F7.3,1H)))
```

```
9013  FORMAT (1H1,15X,*DATA FOR F-COMPARISONS OF MEAN SQUARED ERROR*//
     +     13X,*INTERVAL   NUMBER OF*,18X,*MEAN SQUARED   UNBIASED*/
     +     * PROCEDURE    NUMBER    OBSERVATIONS   MEAN ERROR*,6X,
     +     *ERROR*,7X,*VARIANCE*)
9014  FORMAT (/1X,A9,5X,I2,6X,I7,6X,F9.3,5X,F11.2,3X,F11.2)
9015  FORMAT (/1X,A9,4X,*ALL*,6X,I7,6X,F9.3,5X,F11.2,3X,F11.2)
C     ***************************************************************
C     READ INPUT DATA CARDS
C     ***************************************************************
      NLIMIT = 0
      ITL = 0
100   CONTINUE
      READ (5,9000) MARGIN,VALUE
      IF (EOF(5).EQ.1) GO TO 200
101   CONTINUE
      WRITE (6,9001) MARGIN,VALUE
      IF (MARGIN(1).EQ.8HTIMELINE) ITL=1
      IF (MARGIN(1).EQ.8HINTERVAL) TINT = VALUE
      IF (MARGIN(1).EQ.5HDELAY) TDEL = VALUE
      IF (MARGIN(1).NE.5HLIMIT) GO TO 110
      NLIMIT = NLIMIT + 1
      IF (NLIMIT.LE.5) GO TO 105
      WRITE (6,9011)
      GO TO 100
105   CONTINUE
      TLIM(NLIMIT) = VALUE
      ITLIM(NLIMIT) = VALUE
      GO TO 100
110   CONTINUE
      IF (MARGIN(1).EQ.5HSTART) ASTART = VALUE
      IF (MARGIN(1).EQ.3HEND) AEND = VALUE
      IF (MARGIN(1).EQ.10HSIGNAL OFF) OFFSET = VALUE
      IF (MARGIN(1).EQ.10HSIGNAL FRE.AND.MARGIN(2).EQ.2H01) SF(1)=VALUE
      IF (MARGIN(1).EQ.10HSIGNAL FRE.AND.MARGIN(2).EQ.2H02) SF(2)=VALUE
      IF (MARGIN(1).EQ.10HSIGNAL FRE.AND.MARGIN(2).EQ.2H03) SF(3)=VALUE
      IF (MARGIN(1).EQ.10HSIGNAL AMP.AND.MARGIN(2).EQ.1H1) AMP(1)=VALUE
      IF (MARGIN(1).EQ.10HSIGNAL AMP.AND.MARGIN(2).EQ.1H2) AMP(2)=VALUE
```

```
      IF (MARGIN(1).EQ.10HSIGNAL AMP.AND.MARGIN(2).EQ.1H3) AMP(3)=VALUE
      GO TO 100
C     ****************************************************************
C     MAKE SURE THAT SIGNAL CHARACTERISTICS HAVE BEEN SPECIFIED.
C     ****************************************************************
200   CONTINUE
      X = OFFSET*SF(1)*SF(2)*SF(3)*AMP(1)*AMP(2)*AMP(3)
      IF (X.NE.0.) GO TO 210
      WRITE (6,9002)
      STOP 1
C     ****************************************************************
C     INITIALIZE PROCESSING INDICATORS
C     ****************************************************************
210   CONTINUE
      IMOVE = 0
      CLEVEL = OFFSET
      NINT = 1
C     ****************************************************************
C     READ NEXT MESSAGE ON TAPE
C     ****************************************************************
220   CONTINUE
      TLAST = ST
      IF (ISUB.NE.1) GO TO 2210
      ISUB = 0
      ST = TST
      IM = JM
      ID = JD
      VAL = TVALUE
      GO TO 2201
2210  CONTINUE
      CALL TREAD(ST,IM,ID,VAL)
2201  CONTINUE
      IF (IM.EQ.IEF) GO TO 700
      IF (ST.GT.AEND) GO TO 700

      IF (.NOT.(ST.GE.ASTART.AND.TLAST.LT.ASTART)) GO TO 221
      PRINT = SECT = 0.
      CAFRRT = SAFRRT = CAFSEC = SAFSEC = 0.
      NAFRRT = NAFSEC = 0
```

```
221   CONTINUE
      IF ((ST-CTINT).LT.TINT) GO TO 2220
      TST = ST
      JM = IM
      JD = ID
      TVALUE = VAL
      ST = CTINT + TINT
      IM = ID = 0
      ISUB = 1
2220  CONTINUE
      IF ((IM.NE.13.AND.IM.NE.62).OR.ID.NE.14) GO TO 300
C     ******************************************************************
C     PROCESS MESSAGE THAT INDICATES THE END OF A PROCEDURE.
C     ******************************************************************
      IF (ST.LT.ASTART) GO TO 260
      IF (IMOVE.GT.0) GO TO 240
      IF ((ST-CTINT).EQ.0.) GO TO 260
      IF (NINT.GT.0) NINT = 0
      IF (NINT.LT.0) NINT = -30
      CALL EDATA(PCINT,ST,CLEVEL,CLEVEL,NINT,0)
      IF (ITL.EQ.0) GO TO 260
      TSP = CTINT - TPRIM
      IF (NINT.LT.0) TSP = CTINT - TSECO
      RMSE = SQRT(CUECI/NCI)
      DO 222 I=1,NLIMIT
         PTOT(I) = 100.*TOTCI(I)/NCI
222      CONTINUE
      NTLINE = NTLINE + 2
      IF (NTLINE.LE.60) GO TO 225
      WRITE (6,9003) ITLIM
      NTLINE = 6
225   CONTINUE
      WRITE (6,9004) CTINT,ST,TSR,ERRET,MOVECI,RMSE,(PTOT(I),I=1,NLIMIT)
      GO TO 260
240   CONTINUE
      IBUF = IBUF + 1
      IF (IBUF.LE.10) GO TO 245
      WRITE (6,9005) ST,IM,ID,VAL
      GO TO 260
```

```
245   CONTINUE
      BTIM(IBUF) = PCINT
      FTIM(IBUF) = ST
      INTEND(IBUF) = 0
      IINT(IBUF) = NINT
      BUFINIT(IBUF) = 1.
260   CONTINUE
      CTINT = PCINT = ST
      IF (IMOVE.GT.0) GO TO 262
      MOVECT = 0
      CUFCT = 0.
      NCT = 0

      ECT = 0.
      DO 261 I=1.5
         TOTCI(I) = 0.
261      CONTINUE
262   CONTINUE
      ERRET = CLEVEL - SIGNAL(ST)
      IF (IM.EQ.62) GO TO 270
      NINT = -1
      PRIMT = PRIMT + ST - TPRIM
      TPRIM = 0.
      TSECO = ST
      CAESEC = CAESEC + ABS(ERRET)
      SAESEC = SAESEC + ERRET*ERRET
      NAESEC = NAESEC + 1
      GO TO 220
270   CONTINUE
      TPRIM = ST
      CAERRT = CAERRT + ABS(ERRET)
      SAERRT = SAERRT + ERRET*ERRET
      NAERRT = NAERRT + 1
      NINT = 0
      IF (TDEL.EQ.0.) NINT=1
      SECT = SECT + ST - TSECO
      TSECO = 0.
      GO TO 220
300   CONTINUE
C ---************************************************************************
```

```
C     **********************************************************************
C     IS IT THE END OF A TIMED INTERVALS
C     **********************************************************************
      IF (NINT.LT.0) GO TO 305
      IF ((ST-TPRIM).LT.TDEL) GO TO 400
      IF (CTINT.EQ.TPRIM.AND.TDEL.NE.0.) GO TO 310
305   CONTINUE
      IF ((ST-CTINT).LT.TINT) GO TO 400
310   CONTINUE
      IF (ST.LT.ASTART) GO TO 360
      IF (IMOVE.GT.0) GO TO 340
      CALL EDATA(PCINT,ST,CLEVEL,CLEVEL,NINT,1)
      IF (ITL.EQ.0) GO TO 360
      TSR = CTINT - TPRIM
      IF (NINT.LT.0) TSR = CTINT - TSECO
      RMSE = SQRT(CUECI/NCI)
      DO 311 I=1,NLIMIT
         PTOT(I) = 100.*TOTCI(I)/NCI
311      CONTINUE
      NTLINE = NTLINE + 2
      IF (NTLINE.LE.60) GO TO 320
      WRITE (6,9003) ITLIM
      NTLINE = 6
320   CONTINUE
      WRITE (6,9004) CTINT,ST,TSR,ERRET,MOVECI,RMSE,(PTOT(I),I=1,NLIMIT)
      GO TO 360
340   CONTINUE
      IBUF = IBUF + 1
      IF (IBUF.LE.10) GO TO 345

      WRITE (6,9005) ST,IM,ID,VAL
      GO TO 360
345   CONTINUE
      BTIM(IBUF) = PCINT
      ETIM(IBUF) = ST
      INTEND(IBUF) = 1
      BUFTSR(IBUF) = CTINT - TPRIM
      IF (NINT.LT.0) BUFTSR(IBUF) = CTINT - TSECO
      BUFFAR(IBUF) = ERRET
      BUFCTI(IBUF) = CTINT
```

```
      TINT(IBUF) = NINT
      BUFINIT(IBUF) = 1.
360   CONTINUE
      IF (NINT.LT.0) GO TO 365
      NINT = NINT + 1
      IF (ST.LT.(TPRIM+TDEL+TINT)) NINT=0
      IF (NINT.GT.20) NINT=0
      GO TO 366
365   CONTINUE
      NINT = NINT - 1
      IF (NINT.LT.-20) NINT = -30
366   CONTINUE
      CTINT = PCINT = ST
      IF (IMOVE.GT.0) GO TO 400
      MOVECT = 0
      CUECT = 0.
      NCT = 0
      FCT = 0.
      DO 361 I=1,5
         TOTCI(I) = 0.
361      CONTINUE
400   CONTINUE
C     *************************************************************
C     IS IT THE BEGINNING OF A KNOB TURN&
C     *************************************************************
      IF (IM.NE.35.OR.ID.NE.8) GO TO 500
      IF (ST.LT.ASTART) GO TO 410
      MOVECT = MOVECT + 1
      MOVEALL = MOVEALL + 1
      IF ((ST-PCINT).EQ.0.) GO TO 410
      CALL EDATA(PCINT,ST,CLEVEL,CLEVEL,NINT,0)
410   CONTINUE
      TBMOVE = PCINT = ST
      IMOVE = 1
      IBUF = 0
      GO TO 220
500   CONTINUE
C     *************************************************************
C     IS IT THE END OF A KNOB TURN&
      *************************************************************
```

```
C     *******************************************************************
      IF (IM.NE.36.OR.ID.NE.8) GO TO 600
      IBUF = IBUF + 1
      IF (IBUF.LE.10) GO TO 510
      WRITE (6,9005) ST,IM,ID,VAL
      GO TO 520
510   CONTINUE
      BTIM(IBUF) = PCINT
      ETIM(IBUF) = ST
      INTEND(IBUF) = 0
      IINT(IBUF) = NINT
      BUFINIT(IBUF) = 0.
520   CONTINUE
      IMOVE = 2
      TEMOVE = PCINT - ST
      GO TO 220
600   CONTINUE
      IF (IMOVE.NE.2) GO TO 220
C     *******************************************************************
C     IS THE ACTUAL VALUE OF TRACK CHANGED&
C     *******************************************************************
      IF (IM.NE.80.OR.ID.NE.3) GO TO 220
      IF (IBUF.GT.0) GO TO 610
      WRITE (6,9006) ST,IM,ID,VAL
      GO TO 650
C     *******************************************************************
C     STORE DATA FOR EVENTS IN MOVEMENT BUFFER
C     *******************************************************************
610   CONTINUE
      NBUF = IBUF
C     *******************************************************************
C     DETERMINE PARAMETERS OF RESPONSE-TIME FUNCTION  R(T) = A*T+B
C     *******************************************************************
      A = (VAL - CLEVEL) / (TEMOVE - TBMOVE)
      B = (CLEVEL*TEMOVE - VAL*TBMOVE) / (TEMOVE - TBMOVE)
      IBUF = 0
620   CONTINUE
      IBUF = IBUF + 1
      IF (IBUF.GT.NBUF) GO TO 650
      IF (BTIM(IBUF).LT.ASTART) GO TO 640
```

```
      IF (RTIM(IBUF).EQ.FTIM(IBUF)) GO TO 630
      V1 = A*RTIM(IBUF) + B
      V2 = A*FTIM(IBUF) + B
      CALL EDATA(RTIM(IBUF),FTIM(IBUF),V1,V2,IINT(IBUF),INTEND(IBUF))
630   CONTINUE
      IF (ITL.EQ.0) GO TO 640
      IF (INTEND(IBUF).NE.1) GO TO 640
      TSR = BUFTSR(IBUF)
      DO 631 I=1,NLIMIT
          PTOT(I) = 100.*TOTCI(I)/NCI
631       CONTINUE
      RMSE = SQRT(CUECI/NCI)
      T1 = BUFCTI(IBUF)
      T2 = FTIM(IBUF)
      ERR = BUFEAR(IBUF)
      NTLINE = NTLINE + 2
      IF (NTLINE.LE.60) GO TO 635
      WRITE (6,9003) ITLIM
      NTLINE = 6
635   CONTINUE
      WRITE (6,9004) T1,T2,TSR,ERR,MOVECI,RMSE,(PTOT(I),I=1,NLIMIT)

640   CONTINUE
      IF (BUFINIT(IBUF).EQ.0.) GO TO 620
      MOVECI = 0
      CUECI = 0.
      NCI = 0
      ECI = 0.
      DO 641 I=1,5
          TOTCI(I) = 0.
641       CONTINUE
      GO TO 620
650   CONTINUE
      IBUF = NBUF = 0
      CLEVEL = VAL
      IMOVE = 0
      GO TO 220
C     ********************************************************
```

```
      *************************************************************
C     PROCESSING OF TAPE IS COMPLETE
C     *************************************************************
700   CONTINUE
C     *************************************************************
C     PROCESS FINAL SEGMENT OF DATA
C     *************************************************************
      IF (IM.EQ.IEE) ST = TLAST
      IF (IMOVE.NE.0) GO TO 701
      IF (PCINT.EQ.ST) GO TO 701
      CALL EDATA(PCINT,ST,CLEVEL,CLEVEL,NINT,0)
701   CONTINUE
      IF (NINT.GE.0) PRIMT = PRIMT + TPRIM - ST
      IF (NINT.LT.0) SECT = SECT + TSECO - ST
C     *************************************************************
C     PRINT CUMULATIVE DATA.
C     *************************************************************
      NAME = 7HPRIMARY
      WRITE (6,9007) NAME,ITLIM
      DO 720 N=1,20
         IF (NOBS(N).EQ.0) GO TO 730
         TSR = TDEL + (N-1.)*TINT
         RMSM = SCENI(N)/NOBS(N)
         IF (NOBS(N).EQ.1) GO TO 705
         VAR = SSCENI(N)/(NOBS(N)-1.) - NOBS(N)*RMSM*RMSM/(NOBS(N)-1.)
         IF (VAR.LE.0.) GO TO 705
         RMSSD = SQRT(VAR)
         GO TO 706
705      CONTINUE
         RMSSD = 0.
706      CONTINUE
         XMVSD = 0.
         XMV = SMOVEI(N)/NOBS(N)
         IF (NOBS(N).EQ.1) GO TO 707
         VAR = SSMOVEI(N)/(NOBS(N)-1.) - NOBS(N)*XMV*XMV/(NOBS(N)-1.)
         IF (VAR.LE.0.) GO TO 707
         XMVSD = SQRT(VAR)
707      CONTINUE
         DO 715 I=1,NLIMIT
```

```
            PTOT(I) = STTNI(N,I)/NOBS(N)

              IF (NOBS(N).EQ.1) GO TO 714
              VAR = SSTTNI(N,I)/(NOBS(N)-1.)
     +              - NOBS(N)*PTOT(I)*PTOT(I)/(NOBS(N)-1.)
              IF (VAR.LE.0.) GO TO 714
              PTOTSD(I) = SQRT(VAR)
              GO TO 715
  714         CONTINUE
              PTOTSD(I) = 0.
  715         CONTINUE
          WRITE (6,9008) N,NOBS(N),TSR,XMV,RMSM,(PTOT(I),I=1,NLIMIT)
          WRITE (6,9012) XMVSD,RMSSD,(PTOTSD(I),I=1,NLIMIT)
  720     CONTINUE
  730 CONTINUE
      NAME = 9HSECONDARY
      WRITE (6,9007) NAME,ITLIM
      DO 733 N=1,20
          IF (NSEC(N).EQ.0) GO TO 734
          TSR = (N-1.)*TINT
          RMSSD = 0.
          RMSM = SESEC(N)/NSEC(N)
          IF (NSEC(N).EQ.1) GO TO 731
          VAR = SSESEC(N)/(NSEC(N)-1.)-NSEC(N)*RMSM*RMSM/(NSEC(N)-1.)
          IF (VAR.LE.0.) GO TO 731
          RMSSD = SQRT(VAR)
  731     CONTINUE
          XMV = XMVSD = 0.
          DO 732 I=1,NLIMIT
              PTOT(I) = STTSEC(N,I)/NSEC(N)
              PTOTSD(I) = 0.
              IF (NSEC(N).EQ.1) GO TO 732
              VAR = SSTTSEC(N,I)/(NSEC(N)-1.)
     +              - NSEC(N)*PTOT(I)*PTOT(I)/(NSEC(N)-1.)
              IF (VAR.LE.0.) GO TO 732
              PTOTSD(I) = SQRT(VAR)
  732         CONTINUE
          WRITE (6,9008) N,NSEC(N),TSR,XMV,RMSM,(PTOT(I),I=1,NLIMIT)
          WRITE (6,9012) XMVSD,RMSSD,(PTOTSD(I),I=1,NLIMIT)
  733     CONTINUE
```

```
734   CONTINUE
      WRITE (6,9009) ITLIM
      NAME = 7HPRIMARY
      EREM = CAERRT/NAERRT
      VAR = SAERRT/(NAERRT-1.) - NAERRT*EREM*EREM/(NAERRT-1.)
      ERESD = SQRT(VAR)
      RMSE = SQRT(CUEPRI/NCPRI)
      DO 735 I=1,NLIMIT
         PTOT(I) = 100.*TOTPRI(I)/NCPRI
735      CONTINUE
      WRITE (6,9010) NAME,PRIMT,EREM,ERESD,MOVEALL,RMSE,
     +     (PTOT(I),I=1,NLIMIT)
      IF (NAESEC.LE.1) GO TO 741
      NAME = 9HSECONDARY
      EREM = CAESEC/NAESEC
      VAR = SAESEC/(NAESEC-1.) - NAESEC*EREM*EREM/(NAESEC-1.)
      ERESD = SQRT(VAR)


      IF (NCSEC.EQ.0) GO TO 741
      RMSE = SQRT(CUESEC/NCSEC)
      DO 740 I=1,NLIMIT
         PTOT(I) = 100.*TOTSEC(I)/NCSEC
740      CONTINUE
      IZ = 0
      WRITE (6,9010) NAME,SECT,EREM,ERESD,IZ,RMSE,(PTOT(I),I=1,NLIMIT)
741   CONTINUE
      NAME = 4HBOTH
      T = PRIMT + SECT
      N = NAERRT + NAESEC
      EREM = (CAERRT+CAESEC)/N
      VAR = (SAERRT+SAESEC)/(N-1.) - N*EREM*EREM/(N-1.)
      ERESD = SQRT(VAR)
      RMSE = SQRT((CUEPRI+CUESEC)/(NCPRI+NCSEC))
      DO 745 I=1,NLIMIT
         PTOT(I) = 100.*(TOTPRI(I)+TOTSEC(I))/(NCPRI+NCSEC)
745      CONTINUE
      WRITE (6,9010) NAME,T,EREM,ERESD,MOVEALL,RMSE,(PTOT(I),I=1,NLIMIT)
C     ****************************************************************
C     INITIALIZE ALL VARIABLES IN ORDER TO PROCESS NEXT SEGMENT OF TAPE
      ****************************************************************
```

```
C     ----************************************************************
      CUFCI = CUFPRI = CUFSEC = FCI = 0.
      NCI = NCPRI = NCSEC = MOVECI = 0
      FCUM(1) = FCUM(2) = 0.
      DO 790 I=1,20
         SCENI(I) = SSCENI(I) = SMOVEI(I) = SSMOVEI(I) = 0.
         SESEC(I) = SSESEC(I) = 0.
         NOBS(I) = NSEC(I) = 0
         FCUMI(1,I) = FCUMI(2,I) = 0.
         NIERR(1,I) = NIERR(2,I) = 0
         DO 780 J=1,5
            STTNI(I,J) = SSTTNI(I,J) = STTSEC(I,J) = SSTTSEC(I,J)=0.
780         CONTINUE
790      CONTINUE
      DO 795 I=1,5
         TOTCI(I) = TOTPRI(I) = TOTSEC(I) = 0.
795      CONTINUE
      PRIMT = SECT = 0.
      MOVECI = MOVEALL = 0
      CAFRRT = SAFRRT = CAFSEC = SAESEC = 0.
      NAFRPT = NAESEC = 0
      READ (5,9000) MARGIN,VALUE
      IF (EOF(5).NE.1) GO TO 101
      STOP
      END

      SUBROUTINE EDATA(T1,T2,V1,V2,NN,INTCUM)
      COMMON /STORE/CUFCI,TOTCI(5),NCI,CUFPRI,TOTPRI(5),NCPRI,CUFSEC,
     +   TOTSEC(5),NCSEC,SCENI(20),SSCENI(20),STTNI(20,5),SSTTNI(20,5),
     +   NOBS(20),TLIM(5),NLIMIT,MOVECI,SMOVEI(20),SSMOVEI(20)
     +   ,SESEC(20),SSESEC(20),STTSEC(20,5),SSTTSEC(20,5),NSEC(20)
     +   ,FCUM(2),FCUMI(2,20),NIERR(2,20),FCI
      DATA T/0./

9000  FORMAT (1X,*$$$ ERROR ON CALL TO EDATA*/* T1=*,F8.2,4X,*T2=*,
     +   F8.2,4X,*V1=*,F8.2,4X,*V2=*,F8.2,4X,*NN=*,I3,4X,*INTCUM=*,I2)

      IF (T1.EQ.T2) GO TO 100
      A = (V2 - V1) / (T2 - T1)
      B = (V1*T2 - V2*T1) / (T2 - T1)
```

```
10    CONTINUE
      IF (T.LT.T1) GO TO 20
         ERR = A*T + B - SIGNAL(T)
         ECI = ECI + ERR
         CUECI = CUECI + ERR*ERR
         ABERR = ABS(ERR)
         DO 11 I=1,NLIMIT
              IF (ABERR.LE.TLIM(I)) TOTCI(I)=TOTCI(I)+1.0
11            CONTINUE
         NCI = NCI + 1
         IF (NN.LT.0) GO TO 15
         CUEPRI = CUEPRI + ERR*ERR
         ECUM(1) = ECUM(1) + ERR
         DO 12 I=1,NLIMIT
              IF (ABERR.LE.TLIM(I)) TOTPRI(I)=TOTPRI(I)+1.0
12            CONTINUE
         NCPRI = NCPRI + 1
         GO TO 20
15       CONTINUE
         CUESEC = CUESEC + ERR*ERR
         ECUM(2) = ECUM(2) + ERR
         DO 16 I=1,NLIMIT
              IF (ABERR.LE.TLIM(I)) TOTSEC(I)=TOTSEC(I)+1.0
16            CONTINUE
         NCSEC = NCSEC + 1
20       CONTINUE
         T = T + .01
         IF (T.LT.T2) GO TO 10
100   CONTINUE
      IF (INTCUM.NE.1.OR.NN.EQ.0) RETURN
      IF (NN.LT.0) GO TO 125
      IF (NN.LE.20) GO TO 110
      WRITE (6,9000) T1,T2,V1,V2,NN,INTCUM
      RETURN
110   CONTINUE
      NIERR(1,NN) = NIERR(1,NN) + NCI
      ECUMI(1,NN) = ECUMI(1,NN) + ECI
      SMOVEI(NN) = SMOVEI(NN) + MOVECI
```

```
      SSMOVEI(NN) = SSMOVEI(NN) + MOVECI*MOVECI
      RMSE = SQRT(CUECI/NCI)
      SCENI(NN) = SCENI(NN) + RMSE

      SSCENI(NN) = SSCENI(NN) + RMSE*RMSE
      DO 120 I=1,NLIMIT
          PTOT = 100.*TOTCI(I)/NCI
          STTNI(NN,I) = STTNI(NN,I) + PTOT
          SSTTNI(NN,I) = SSTTNI(NN,I) + PTOT*PTOT
120       CONTINUE
      NOBS(NN) = NOBS(NN) + 1
      RETURN
125   CONTINUE
      MM = -NN
      IF (MM.EQ.30) RETURN
      IF (MM.LE.20) GO TO 130
      WRITE (6,9000) T1,T2,V1,V2,NN,INTCUM
      RETURN
130   CONTINUE
      NIERR(2,MM) = NIERR(2,MM) + NCI
      ECUMI(2,MM) = ECUMI(2,MM) + ECI
      RMSE = SQRT(CUECI/NCI)
      SESEC(MM) = SESEC(MM) + RMSE
      SSESEC(MM) = SSESEC(MM) + RMSE*RMSE
      DO 140 I=1,NLIMIT
          PTOT = 100.*TOTCI(I)/NCI
          STTSEC(MM,I) = STTSEC(MM,I) + PTOT
          SSTTSEC(MM,I) = SSTTSEC(MM,I) + PTOT*PTOT
140       CONTINUE
      NSEC(MM) = NSEC(MM) + 1
      RETURN
      END

      SUBROUTINE TREAD(ST,IM,ID,VAL)
      DIMENSION BUFF(126)
      DATA ICOUNT/126/,IFF/77777777770000000000B/,
     +    FF/77777777777777777777B/
      ICOUNT = ICOUNT + 2
      IF (ICOUNT.LE.125) GO TO 20
```

```
      READ(12) BUFF
      IF (EOF(12).NE.0) GO TO 50
      ICOUNT = 1
20    CONTINUE
      IF (BUFF(ICOUNT).EQ.FF) GO TO 50
      ST = BUFF(ICOUNT).AND.IFF
      IM = BUFF(ICOUNT).AND.7777B
      ID = SHIFT(BUFF(ICOUNT),-12).AND.7777B
      VAL = BUFF(ICOUNT+1)
      RETURN
50    CONTINUE
      IM = IFF
      RETURN
      END
```

```
      FUNCTION SIGNAL(T)                                                SIGNAL       2
      COMMON /SIG/SF(3),AMP(3),OFFSET                                   SIG          2
      SIGNAL = AMP(1)*SIN(SF(1)*T) + AMP(2)*SIN(SF(2)*T)                MAR18A       1
     +          + AMP(3)*SIN(SF(3)*T) + OFFSET                          MAR18A       2
      RETURN                                                            SIGNAL       6
      END                                                               SIGNAL       7
```

TIMELINE	-.000000
START	20.000000
INTERVAL	2.000000
SIGNAL OFFSET	10000.000000
SIGNAL FREQ1	.628318
SIGNAL FREQ2	.897597
SIGNAL FREQ3	1.570795
SIGNAL AMP1	4000.000000
SIGNAL AMP2	2800.000000
SIGNAL AMP3	1600.000000
LIMIT	1600.000000
LIMIT	800.000000
LIMIT	500.000000
LIMIT	300.000000
LIMIT	100.000000

TIMELINE DATA FOR PRIMARY TASK BY TIME INTERVAL

INTERVAL START	INTERVAL END	TIME SINCE RETURN	ERROR AT RETURN	NO. MOVES	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
18.17	20.17	2.00	355.00	0	6216.75	.000	.000	.000	.000	.000
20.17	22.17	.00	1779.54	5	1162.65	85.500	60.500	49.000	25.000	8.500
22.17	24.17	2.00	1779.54	2	392.38	100.000	98.500	76.500	52.000	17.000
24.17	26.17	4.00	1779.54	8	628.01	99.000	83.500	56.000	32.500	9.000
26.17	26.26	6.00	1779.54	0	1650.52	22.222	.000	.000	.000	.000
26.26	28.26	.00	-1586.82	0	2310.81	63.500	40.500	27.500	14.500	.000
28.26	30.26	2.00	-1586.82	0	9438.35	.000	.000	.000	.000	.000
30.26	32.26	.00	-10391.27	3	6358.89	46.000	42.500	41.000	33.500	10.500
32.26	34.26	2.00	-10391.27	4	378.37	100.000	100.000	82.500	38.000	13.000
34.26	36.26	4.00	-10391.27	2	356.33	100.000	100.000	89.000	42.500	13.000
36.26	36.27	6.00	-10391.27	0	203.78	100.000	100.000	100.000	100.000	.000
36.27	38.27	.00	195.60	0	1784.50	67.000	54.500	49.000	44.500	17.000
38.27	40.27	2.00	195.60	0	4779.81	.000	.000	.000	.000	.000
40.27	42.27	.00	2000.23	5	1263.12	69.500	45.500	31.500	13.500	4.000
42.27	44.27	2.00	2000.23	2	461.27	100.000	91.500	65.000	62.500	13.500
44.27	46.27	4.00	2000.23	8	1072.57	84.000	80.000	66.000	44.500	16.500
46.29	48.29	.00	-1909.95	0	1442.84	71.500	52.000	43.500	36.500	12.000
48.29	50.29	2.00	-1909.95	0	8268.82	.000	.000	.000	.000	.000
50.29	52.29	.00	-10641.94	5	6331.20	53.500	23.500	19.500	10.500	2.000
52.29	54.29	2.00	-10641.94	9	1214.11	79.500	38.000	34.000	28.000	11.000
54.29	56.29	4.00	-10641.94	3	616.90	95.500	84.000	69.500	56.000	32.500
56.29	56.32	6.00	-10641.94	0	386.93	100.000	100.000	100.000	.000	.000
56.32	58.32	.00	471.79	0	640.33	100.000	68.500	42.500	26.000	8.000
58.32	60.32	2.00	471.79	0	1325.12	66.000	25.500	18.000	11.000	3.500
60.32	62.32	.00	-604.19	2	859.52	91.000	64.000	59.000	58.500	8.000
62.32	64.32	2.00	-604.19	4	497.27	100.000	90.000	64.500	40.500	7.000
64.32	66.32	4.00	-604.19	7	818.45	92.537	84.080	56.219	26.368	9.453
66.32	66.34	6.00	-604.19	0	2965.88	.000	.000	.000	.000	.000

TIMELINE DATA FOR PRIMARY TASK BY TIME INTERVAL

INTERVAL START	INTERVAL END	TIME SINCE RETURN	ERROR AT RETURN	NO. MOVES	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
66.34	68.34	.00	-2831.30	0	1036.30	89.500	49.500	30.000	17.500	5.500
68.34	70.34	2.00	-2831.30	0	4570.35	11.500	.500	.000	.000	.000
70.34	72.34	.00	-7062.05	6	5694.56	19.500	4.500	3.000	1.500	.500
72.34	74.34	2.00	-7062.05	7	1927.63	77.000	42.000	29.500	19.000	7.500
74.34	76.34	4.00	-7062.05	5	1099.11	82.000	65.500	54.500	33.000	11.000
76.34	76.34	6.00	-7062.05	0	923.37	100.000	.000	.000	.000	.000
76.34	78.34	.00	-828.00	0	2527.82	15.000	.000	.000	.000	.000
78.34	80.34	2.00	-828.00	0	4451.45	.000	.000	.000	.000	.000
80.34	82.34	.00	-6623.24	6	3580.86	58.500	27.000	14.500	6.500	2.000
82.34	84.34	2.00	-6623.24	5	540.78	100.000	86.000	50.000	33.000	12.500
84.34	86.34	4.00	-6623.24	5	664.28	96.500	82.500	57.500	24.500	9.000
86.34	86.36	6.00	-6623.24	0	1160.81	100.000	.000	.000	.000	.000
86.36	88.36	.00	-1133.31	0	1963.76	32.000	18.000	11.000	7.000	2.000
88.36	90.36	2.00	-1133.31	0	310.43	91.000	55.000	39.000	24.500	8.500
90.36	92.36	.00	-1137.29	6	1856.63	42.500	16.000	9.500	2.500	.500
92.36	94.36	2.00	-1137.29	9	802.04	94.000	74.000	52.500	31.500	10.500
94.36	96.36	4.00	-1137.29	4	1144.33	78.000	56.000	20.000	15.000	5.000
96.36	96.37	6.00	-1137.29	0	746.31	100.000	100.000	.000	.000	.000
96.37	98.37	.00	-744.61	0	1551.96	56.281	3.015	.000	.000	.000
98.37	100.37	2.00	-744.61	0	6141.20	.000	.000	.000	.000	.000
100.37	102.37	.00	-9854.89	7	5662.54	43.500	30.500	23.000	17.000	7.000
102.37	104.37	2.00	-9854.89	4	372.70	100.000	99.500	80.000	56.000	12.500
104.37	106.37	4.00	-9854.89	7	525.83	100.000	92.500	51.500	29.000	8.500
106.37	106.46	6.00	-9854.89	0	677.78	100.000	100.000	.000	.000	.000
106.46	108.46	.00	-545.42	0	490.67	100.000	75.000	44.000	25.500	6.000
108.46	110.46	2.00	-545.42	0	415.81	100.000	100.000	66.500	56.000	34.500
110.46	112.46	.00	607.95	6	1269.53	70.500	41.500	30.000	15.500	5.000
112.46	114.46	2.00	607.95	6	1093.62	89.000	38.500	21.000	12.500	5.500

TIMELINE DATA FOR PRIMARY TASK BY TIME INTERVAL

INTERVAL START	INTERVAL END	TIME SINCE RETURN	ERROR AT RETURN	NO. MOVES	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
114.46	116.46	4.00	607.35	6	409.36	100.000	95.000	77.000	49.000	16.500
116.46	116.46	6.00	607.95	0	42.05	100.000	100.000	100.000	100.000	100.000
116.46	118.46	.00	-40.23	0	1225.44	74.000	32.500	21.000	13.500	5.500
118.46	120.46	2.00	-40.23	0	4657.69	39.000	19.500	12.000	7.500	2.500
120.46	122.46	.00	-9195.19	8	5658.80	31.500	13.500	10.000	8.000	4.500
122.46	124.46	2.00	-9195.19	2	853.55	92.500	69.500	39.000	9.000	3.500
124.46	126.46	4.00	-9195.19	2	397.24	100.000	97.000	85.000	44.500	14.500
126.46	126.46	6.00	-9195.19	0	519.65	100.000	100.000	.000	.000	.000
126.46	128.46	.00	512.42	0	1195.68	77.883	63.317	25.628	14.573	4.523
128.46	130.46	2.00	512.42	0	2355.78	28.000	9.000	.000	.000	.000
130.46	132.46	.00	-555.65	5	620.70	100.000	81.500	49.500	18.000	7.500
132.46	134.46	2.00	-555.65	6	615.90	100.000	89.500	57.500	42.500	16.000
134.46	136.46	4.00	-555.65	5	380.13	100.000	99.500	77.500	51.000	16.500
136.46	136.48	6.00	-555.65	0	189.75	100.000	100.000	100.000	100.000	.000
136.48	138.48	.00	201.88	0	3438.26	26.500	12.500	6.500	2.000	.000
138.48	140.48	2.00	201.88	0	3544.16	22.000	11.500	7.000	4.000	1.500
140.48	142.48	.00	-5272.12	8	3928.42	39.801	32.836	25.373	17.910	8.458
142.48	144.48	2.00	-5272.12	4	760.50	93.467	76.884	51.256	33.668	13.568
144.48	146.48	4.00	-5272.12	7	349.25	100.000	100.000	87.000	51.000	16.000
146.48	146.50	6.00	-5272.12	0	354.90	100.000	100.000	100.000	.000	.000
146.50	148.50	.00	-341.59	0	3267.67	44.500	31.500	24.000	13.500	.000
148.50	150.50	2.00	-341.59	0	7421.87	.000	.000	.000	.000	.000
150.50	152.50	.00	-6026.16	7	3011.83	68.000	51.500	40.500	30.000	13.000
152.50	154.50	2.00	-6026.16	2	322.19	100.000	100.000	83.500	67.000	39.000
154.50	156.50	4.00	-6026.16	2	396.56	100.000	100.000	84.000	18.000	3.000
156.50	156.52	6.00	-6026.16	0	63.09	100.000	100.000	100.000	100.000	100.000
156.52	158.52	.00	-50.55	0	3752.88	33.500	21.000	15.000	10.500	5.000
158.52	160.52	2.00	-50.55	0	4925.11	21.500	12.000	7.500	4.500	1.500

TIMELINE DATA FOR PRIMARY TASK BY TIME INTERVAL

INTERVAL START	INTERVAL END	TIME SINCE RETURN	ERROR AT RETURN	NO. MOVES	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
160.52	162.52	.00	-1268.09	7	1971.05	55.000	15.500	6.000	3.000	.500
162.52	164.52	2.00	-1268.09	4	352.76	100.000	100.000	81.500	54.000	18.000
164.52	166.52	4.00	-1268.09	9	748.27	98.500	63.500	42.000	26.500	5.000
166.52	166.55	6.00	-1268.09	0	1221.83	100.000	.000	.000	.000	.000
166.55	168.55	.00	-1198.42	0	3819.80	40.500	3.000	.000	.000	.000
168.55	170.55	2.00	-1198.42	0	10497.26	.000	.000	.000	.000	.000
170.55	172.55	.00	-10676.01	3	6327.70	52.500	50.500	35.000	22.500	4.500
172.55	174.55	2.00	-10676.01	5	413.23	100.000	93.500	76.500	57.000	19.000
174.55	176.55	4.00	-10676.01	2	327.78	100.000	97.500	88.000	73.500	24.000
176.55	176.56	6.00	-10676.01	0	129.52	100.000	100.000	100.000	100.000	.000
176.56	178.56	.00	-126.75	0	2590.53	52.261	40.201	34.673	30.151	8.040
178.56	180.56	2.00	-126.75	0	4487.38	10.000	1.500	.000	.000	.000
180.56	182.56	.00	642.01	3	857.96	95.000	62.000	22.000	10.500	3.500
182.56	184.56	2.00	642.01	0	322.84	100.000	100.000	100.000	48.000	16.000
184.56	186.56	4.00	642.01	9	797.79	100.000	62.000	31.000	17.000	3.000
186.56	186.59	6.00	642.01	0	69.46	100.000	100.000	100.000	100.000	66.667
186.59	188.59	.00	10.07	0	1620.66	77.500	32.500	19.000	11.500	3.500
188.59	190.59	2.00	10.07	0	7987.92	.000	.000	.000	.000	.000
190.59	192.59	.00	-9873.67	4	6783.77	36.500	33.000	27.000	2.500	1.000
192.59	194.59	2.00	-9873.67	10	1133.65	81.095	71.642	52.736	35.323	11.940
194.59	196.59	4.00	-9873.67	5	740.34	94.975	76.884	63.317	38.693	14.070
196.59	196.60	6.00	-9873.67	0	301.80	100.000	100.000	100.000	.000	.000
196.60	198.60	.00	284.13	0	922.21	90.000	72.500	65.500	36.500	11.500
198.60	200.60	2.00	284.13	0	1708.47	44.500	29.000	18.000	10.500	3.500
200.60	202.60	.00	-792.51	2	865.03	[illegible]	59.500	57.500	56.500	37.500
202.60	204.60	2.00	-792.51	4	323.99	100.000	100.000	85.500	61.500	14.000
204.60	206.60	4.00	-792.51	7	1094.97	81.000	70.500	54.500	35.000	9.000
206.60	206.61	6.00	-792.51	0	1687.34	.000	.000	.000	.000	.000

TIMELINE DATA FOR PRIMARY TASK BY TIME INTERVAL

INTERVAL START	INTERVAL END	TIME SINCE RETURN	ERROR AT RETURN	NO. MOVES	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
206.61	208.61	.00	-1658.75	0	879.47	91.000	72.000	39.000	21.500	7.000
209.61	210.61	2.00	-1658.75	0	5746.44	.000	.000	.000	.000	.000
210.61	212.61	.00	-8280.25	7	6303.68	14.500	7.500	4.000	2.500	1.000
212.61	214.61	2.00	-8280.25	7	929.11	91.000	85.000	72.500	52.000	18.000
214.61	216.61	4.00	-8280.25	5	1277.92	80.000	65.500	47.500	27.500	8.000
216.61	216.64	6.00	-8280.25	0	269.01	100.000	100.000	100.000	100.000	.000

CUMULATIVE DATA FOR PRIMARY TASK BY TIME INTERVAL

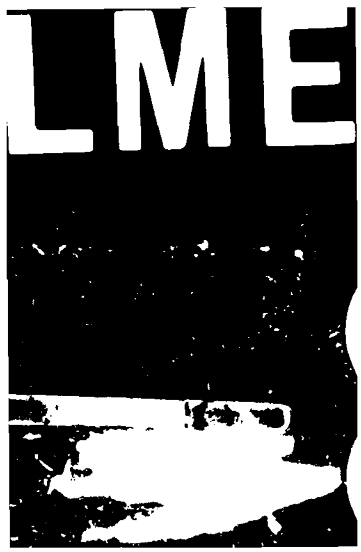
INTERVAL NUMBER	NUMBER OF OBS.	TIME SINCE RETURN	NUMBER OF MOVEMENTS	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
1	20	.00	5.25	3523.77	58.165	38.142	27.844	17.771	6.473
			(1.89)	(2374.44)	(25.076)	(21.302)	(17.471)	(16.488)	(8.160)
2	20	2.00	4.80	680.40	94.878	82.201	62.775	41.650	13.950
			(2.65)	(416.03)	(7.674)	(20.960)	(21.304)	(16.594)	(7.270)
3	20	4.00	5.40	692.27	94.101	82.773	62.852	36.753	12.176
			(2.35)	(308.29)	(8.097)	(14.773)	(19.487)	(14.855)	(7.162)

CUMULATIVE DATA FOR SECONDARY TASK BY TIME INTERVAL

INTERVAL NUMBER	NUMBER OF OBS.	TIME SINCE RETURN	NUMBER OF MOVEMENTS	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
1	19	.00	.00	1914.82	63.286	39.054	26.200	17.117	5.030
			(.00)	(1050.61)	(25.895)	(24.958)	(18.388)	(13.259)	(4.833)
2	19	2.00	.00	5131.28	21.342	13.553	8.842	6.211	2.921
			(.00)	(2823.78)	(32.199)	(25.846)	(17.248)	(13.596)	(7.943)

CUMULATIVE DATA BY PROCEDURE

PROCEDURE	TOTAL TIME	ABSOLUTE ERROR AT ENTRY MEAN	ABSOLUTE ERROR AT ENTRY S.D.	NO. MOVES	RMS ERROR	PERCENT TOT(1600)	PERCENT TOT(800)	PERCENT TOT(500)	PERCENT TOT(300)	PERCENT TOT(100)
PRIMARY	120.48	5164.21	4101.02	309	2510.64	82.350	67.662	51.091	32.022	10.854
SECONDARY	80.00	748.65	755.85	0	4353.37	41.939	25.856	17.071	11.363	3.873
BOTH	200.48	2956.43	3670.31	309	3357.39	66.475	51.239	37.727	23.907	8.112


LME